Link Layer (LL)

Bluetooth® Test Suite

- **Revision**: LL.TS.p17
- **Revision Date**: 2020-01-07
- **Group Prepared By**: BTI
- **Feedback Email**: bti-main@bluetooth.org
## Contents

1. **Scope** ................................................................. 17

2. **References, Definitions, and Abbreviations** ................................................................. 18

2.1 References ........................................................................ 18

3. **Test Suite Structure (TSS)** ................................................................. 19

3.1 Test Strategy ........................................................................ 19

3.2 Test Groups ........................................................................ 21

3.2.1 Device Discovery ................................................................. 21

3.2.2 Connection Handling ................................................................. 21

3.2.3 Timing ........................................................................ 21

3.2.4 Radio Frame Encoding ................................................................. 21

3.2.5 Frequency Hopping ................................................................. 21

3.2.6 Packet Formats ....................................................................... 21

3.2.7 Security ........................................................................ 21

3.2.8 Connected Isochronous Stream ................................................................. 21

3.2.9 Broadcast Isochronous Stream ................................................................. 22

3.2.10 Isochronous Test Mode ................................................................. 22

3.2.11 Power Control ....................................................................... 22

3.3 Behavior Tests ........................................................................ 22

3.3.1 Valid Behavior (BV) Tests ................................................................. 22

3.3.2 Invalid Behavior (BI) Tests ................................................................. 22

3.4 Test Realization ...................................................................... 22

4. **Test Cases (TC)** ................................................................. 23

4.1 Introduction ........................................................................ 23

4.1.1 Test Case Naming Conventions ................................................................. 23

4.1.2 Conformance ........................................................................ 24

4.1.3 Common Packet Contents ................................................................. 24

4.1.3.1 Access Address Used ................................................................. 24

4.1.4 Common Message Contents ................................................................. 25

4.1.4.1 Checksums Used ....................................................................... 25

4.1.4.2 Device Addresses Used ................................................................. 25

4.1.5 Common Initial and Final Conditions ................................................................. 25

4.1.5.1 ADV ............................................................................... 30

4.1.5.2 SCN ............................................................................... 37

4.1.5.3 INI ............................................................................... 39

4.1.5.4 SLA ............................................................................... 40

4.1.5.5 MAS ............................................................................... 43

4.1.5.6 SEC ............................................................................... 46

4.1.5.7 Common Test Procedure Steps ................................................................. 48

4.1.5.8 Pass/Inconclusive/Fail Verdict Conventions ................................................................. 58

4.1.5.9 ISO ............................................................................... 58

4.1.6 Data Fragmentation over HCI ................................................................. 62

4.1.7 Data PDUs and Empty PDUs ................................................................. 63

4.1.8 Outstanding Commands Prior to Disconnection ................................................................. 63

4.2 DDI ............................................................................... 63

4.2.1 Common PDU Contents ................................................................. 63

4.2.1.1 Legacy Advertising ................................................................. 63
4.2.2.41 LL/DDI/ADV/BV-62-C [Periodic Advertising, Channel Map Update] .................................................... 170
4.2.3 SCN .................................................................................................................................................. 172
4.2.3.1 Common PDU Contents .................................................................................................................. 172
4.2.3.2 LL/DDI/SCN/BV-01-C [Passive Scanning; Non-connectable] .......................................................... 172
4.2.2.39 LL/DDI/ADV/BV-03-C [Advertising Data: Non-Connectable] ......................................................... 69
4.2.2.38 LL/DDI/ADV/BV-04-C [Advertising Data: Undirected] ................................................................. 71
4.2.2.37 LL/DDI/ADV/BV-02-C [Undirected Advertising Events] ............................................................ 67
4.2.2.36 LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events] ..................................................... 65
4.2.2.35 LL/DDI/ADV/BV-00-C [Scan Request: Undirected Connectable] .................................................. 73
4.2.2.34 LL/DDI/ADV/BV-04-C [Scan Request Connection Request] ......................................................... 77
4.2.2.33 LL/DDI/ADV/BV-03-C [Scan Request Device Filtering] ............................................................... 79
4.2.2.32 LL/DDI/ADV/BV-02-C [Scan Request: Discoverable] ................................................................. 91
4.2.2.31 LL/DDI/ADV/BV-01-C [Scan Request Invalid CRC] ................................................................. 96
4.2.2.30 LL/DDI/ADV/BV-20-C [Advertising Always Using the LE 1M PHY] ........................................... 101
4.2.2.29 LL/DDI/ADV/BV-19-C [Low Duty Cycle Directed Advertising Events] ..................................... 99
4.2.2.28 LL/DDI/ADV/BV-18-C [Device Filtering: Discoverable] ............................................................ 93
4.2.2.27 LL/DDI/ADV/BV-17-C [Scan Request: Discoverable] ............................................................... 91
4.2.2.26 LL/DDI/ADV/BV-11-C [Directed Advertising Events] ............................................................... 85
4.2.2.25 LL/DDI/ADV/BV-27-C [Extended Advertising, Host Modifying Data and ADI] ......................... 122
4.2.2.24 LL/DDI/ADV/BV-16-C [Advertising Data: Discoverable] ......................................................... 89
4.2.2.23 LL/DDI/ADV/BV-15-C [Discoverable Advertising Events] ......................................................... 87
4.2.2.22 LL/DDI/ADV/BV-14-C [Discoverable Advertising, Legacy PDUs, Undirected] ......................... 105
4.2.2.21 LL/DDI/ADV/BV-13-C [Discoverable Advertising, Non-Connectable] ..................................... 107
4.2.2.20 LL/DDI/ADV/BV-12-C [Discoverable Advertising Events] ......................................................... 86
4.2.2.19 LL/DDI/ADV/BV-11-C [Directed Advertising Events] ............................................................... 85
4.2.2.18 LL/DDI/ADV/BV-10-C [Directly Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.17 LL/DDI/ADV/BV-09-C [Connection Request Device Filtering] .................................................. 82
4.2.2.16 LL/DDI/ADV/BV-08-C [Scan Request Device Filtering] ............................................................ 79
4.2.2.15 LL/DDI/ADV/BV-07-C [Scan Request Connection Request] ....................................................... 77
4.2.2.14 LL/DDI/ADV/BV-06-C [Connection Request] ............................................................................. 75
4.2.2.13 LL/DDI/ADV/BV-05-C [Scan Request: Undirected Connectable] .............................................. 73
4.2.2.12 LL/DDI/ADV/BV-04-C [Scan Request Connection Request] ....................................................... 77
4.2.2.11 LL/DDI/ADV/BV-03-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.10 LL/DDI/ADV/BV-02-C [Scan Request: Discoverable] ............................................................... 91
4.2.2.9 LL/DDI/ADV/BV-01-C [Scan Request Invalid CRC] ................................................................. 96
4.2.2.8 LL/DDI/ADV/BV-00-C [Scan Request: Discoverable] ............................................................... 91
4.2.2.7 LL/DDI/ADV/BV-01-C [Scan Request Connection Request] ....................................................... 77
4.2.2.6 LL/DDI/ADV/BV-00-C [Scan Request Device Filtering] ............................................................ 79
4.2.2.5 LL/DDI/ADV/BV-04-C [Scan Request Connection Request] ....................................................... 77
4.2.2.4 LL/DDI/ADV/BV-03-C [Directed Advertising, Multiple Sets, Single PHY, LE 1M PHY] .......... 128
4.2.2.3 LL/DDI/ADV/BV-02-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHY] .......... 132
4.2.2.2 LL/DDI/ADV/BV-01-C [Scan Request Invalid CRC] ................................................................. 96
4.2.2.1 LL/DDI/ADV/BV-00-C [Scan Request: Discoverable] ............................................................... 91
4.2.2.38 LL/DDI/ADV/BV-05-C [Scan Request: Undirected Connectable] .............................................. 73
4.2.2.37 LL/DDI/ADV/BV-04-C [Scan Request Connection Request] ....................................................... 77
4.2.2.36 LL/DDI/ADV/BV-03-C [Directed Advertising, Multiple Sets, Maximum Supported Sets] .......... 149
4.2.2.35 LL/DDI/ADV/BV-02-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHY] .......... 132
4.2.2.34 LL/DDI/ADV/BV-01-C [Scan Request Invalid CRC] ................................................................. 96
4.2.2.33 LL/DDI/ADV/BV-00-C [Scan Request: Discoverable] ............................................................... 91
4.2.2.32 LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events] ..................................................... 65
4.2.2.31 LL/DDI/ADV/BV-02-C [Directed Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)] ......................................................... 143
4.2.2.30 LL/DDI/ADV/BV-03-C [Directed Advertising, Overlapping Extended Advertising Events] ....... 125
4.2.2.29 LL/DDI/ADV/BV-02-C [Directed Advertising, Multiple Sets, Single PHY, LE 1M PHY] .......... 128
4.2.2.28 LL/DDI/ADV/BV-01-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHY] .......... 132
4.2.2.27 LL/DDI/ADV/BV-00-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHY] .......... 132
4.2.2.26 LL/DDI/ADV/BV-01-C [Directed Advertising, Overlapping Extended Advertising Events] ....... 125
4.2.2.25 LL/DDI/ADV/BV-02-C [Directed Advertising, Host Modifying Data and ADI] ......................... 122
4.2.2.24 LL/DDI/ADV/BV-03-C [Directed Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)] ......................................................... 143
4.2.2.23 LL/DDI/ADV/BV-02-C [Directed Advertising, Multiple Sets, Maximum Supported Sets] .......... 149
4.2.2.22 LL/DDI/ADV/BV-01-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHY] .......... 132
4.2.2.21 LL/DDI/ADV/BV-00-C [Directed Advertising, Multiple Sets, Single PHY, LE Coded PHYs] ....... 135
4.2.2.20 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.19 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.18 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.17 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.16 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.15 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.14 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.13 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.12 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.11 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.10 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.9 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.8 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.7 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.6 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.5 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.4 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.3 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.2 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.1 LL/DDI/ADV/BV-00-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.2.0 LL/DDI/ADV/BV-01-C [Directed Advertising, Legacy PDUs, Non-Connectable] ....................... 103
4.2.1.2 Extended Advertising ..................................................................................................................... 65
4.2.2 ADV .................................................................................................................................................. 65
4.2.2.1 Common PDU Contents .................................................................................................................. 65
### Common PDU Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3.3</td>
<td>LL/DDI/SCN/BV-02-C [Passive Scanning Device Filtering]</td>
</tr>
<tr>
<td>4.2.3.4</td>
<td>LL/DDI/SCN/BV-03-C [Active Scanning]</td>
</tr>
<tr>
<td>4.2.3.5</td>
<td>LL/DDI/SCN/BV-04-C [Active Scanning Device Filtering]</td>
</tr>
<tr>
<td>4.2.3.6</td>
<td>LL/DDI/SCN/BV-05-C [Scanning For Advertiser Types]</td>
</tr>
<tr>
<td>4.2.3.7</td>
<td>LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected Events]</td>
</tr>
<tr>
<td>4.2.3.8</td>
<td>LL/DDI/SCN/BV-11-C [Passive Scanning: Directed Events]</td>
</tr>
<tr>
<td>4.2.3.9</td>
<td>LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable Events]</td>
</tr>
<tr>
<td>4.2.3.10</td>
<td>LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]</td>
</tr>
<tr>
<td>4.2.3.11</td>
<td>LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning; Directed Events to an address different from the scanner's address]</td>
</tr>
<tr>
<td>4.2.3.12</td>
<td>LL/DDI/SCN/BV-15-C [Network Privacy – Active Scanning, no Local IRK, no Peer IRK]</td>
</tr>
<tr>
<td>4.2.3.13</td>
<td>LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK]</td>
</tr>
<tr>
<td>4.2.3.14</td>
<td>LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK]</td>
</tr>
<tr>
<td>4.2.3.15</td>
<td>LL/DDI/SCN/BV-18-C [Network Privacy – Active Scanning, Local IRK, Peer IRK]</td>
</tr>
<tr>
<td>4.2.3.16</td>
<td>Extended Scanning, Passive</td>
</tr>
<tr>
<td>4.2.3.17</td>
<td>Extended Scanning, Active</td>
</tr>
<tr>
<td>4.2.3.18</td>
<td>Extended Scanning, Periodic Advertising Reception</td>
</tr>
<tr>
<td>4.2.3.19</td>
<td>LL/DDI/SCN/BV-23-C [Extended Scanning, Multiple Sets, Passive, Multiple PHYs (All Supported PHYs)]</td>
</tr>
<tr>
<td>4.2.3.20</td>
<td>Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)</td>
</tr>
<tr>
<td>4.2.3.21</td>
<td>LL/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)]</td>
</tr>
<tr>
<td>4.2.3.22</td>
<td>LL/DDI/SCN/BV-26-C [Network Privacy – Passive Scanning, Peer IRK, Ignore Identity Address]</td>
</tr>
<tr>
<td>4.2.3.23</td>
<td>LL/DDI/SCN/BV-28-C [Device Privacy – Passive Scanning, Peer IRK, Accept Identity Address]</td>
</tr>
<tr>
<td>4.2.3.24</td>
<td>AoD Connectionless CTE Scanning</td>
</tr>
<tr>
<td>4.2.3.25</td>
<td>AoA Connectionless CTE Scanning</td>
</tr>
<tr>
<td>4.2.3.26</td>
<td>AoD Connectionless CTE Scanning, Incorrect CRC</td>
</tr>
<tr>
<td>4.2.3.27</td>
<td>AoA Connectionless CTE Scanning, Incorrect CRC</td>
</tr>
<tr>
<td>4.2.3.28</td>
<td>Privacy - Extended Scanning, Active</td>
</tr>
<tr>
<td>4.2.3.29</td>
<td>LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]</td>
</tr>
<tr>
<td>4.2.3.30</td>
<td>LL/DDI/SCN/BV-35-C [Connectionless CTE Scanning, Filter Wrong CTE Types on Synchronization]</td>
</tr>
<tr>
<td>4.2.3.31</td>
<td>LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]</td>
</tr>
<tr>
<td>4.2.3.32</td>
<td>LL/DDI/SCN/BV-37-C [First AUX_SYNC_IND Never Received]</td>
</tr>
<tr>
<td>4.2.3.33</td>
<td>Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled</td>
</tr>
<tr>
<td>4.2.3.34</td>
<td>LL/DDI/SCN/BI-01-C [Active Scanning Invalid CRC]</td>
</tr>
<tr>
<td>4.2.3.35</td>
<td>LL/DDI/SCN/BI-02-C [Passive Scanning Invalid CRC]</td>
</tr>
<tr>
<td>4.2.3.36</td>
<td>LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Wrong AdvA in Response]</td>
</tr>
<tr>
<td>4.2.3.37</td>
<td>LL/DDI/SCN/BV-67-C [Periodic Advertising Reception, Channel Map Update]</td>
</tr>
</tbody>
</table>

### CON

- 4.3.1 Common PDU Contents
- 4.3.2 ADV
- 4.3.2.1 Common PDU Contents
- 4.3.2.2 Common Variables
- 4.3.2.3 LL/CON/ADV/BV-01-C [Accepting Connections]
- 4.3.2.4 LL/CON/ADV/BV-02-C [Accepting Connections Timeout]
- 4.3.2.5 LL/CON/ADV/BV-03-C [Master Missing Slave Packets]
- 4.3.2.6 LL/CON/ADV/BV-04-C [Directed Advertising Connection]
- 4.3.2.7 Extended Advertising, Accepting Connections
4.3.2.8 LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections] ..........278
4.3.2.9 LL/CON/ADV/BV-07-C [Accepting Connections, Channel Selection Algorithm #1] ..........281
4.3.2.10 LL/CON/ADV/BV-08-C [Directed Advertising Connection, Channel Selection Algorithm #1] ......283
4.3.2.11 LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2] ..........285
4.3.2.12 LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2] ......287
4.3.2.13 LL/CON/ADV/BV-11-C [Accepting Connections, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2] ..................................................291
4.3.2.15 Extended Advertising, Accepting Connections with Random address ..................................293
4.3.3 INI .............................................................................296
4.3.3.1 LL/CON/INI/BV-01-C [Connection Initiation] ........................................................................297
4.3.3.2 LL/CON/INI/BV-02-C [Connecting to Directed Advertising] .................................................300
4.3.3.3 LL/CON/INI/BV-03-C [Connection Initiation Missed Replies] ..................................................302
4.3.3.4 LL/CON/INI/BV-04-C [Connection Initiation Timeout] ..............................................................304
4.3.3.5 LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected] .................................................306
4.3.3.6 LL/CON/INI/BV-07-C [Initiation Device Filtering: Directed] ....................................................308
4.3.3.7 LL/CON/INI/BV-08-C [Network Privacy – Connection Establishment responding to connectable undirected advertising, Initiator] ......................................................................................310
4.3.3.8 LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator] ..................................................................................................................312
4.3.3.9 LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator] .................................................................314
4.3.3.10 LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator] ......................................................316
4.3.3.11 LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator] ..........................................................319
4.3.3.12 Extended Scanning, Connection Initiation .............................................................................321
4.3.3.13 LL/CON/INI/BV-14-C [Connection Initiation, Channel Selection Algorithm #1] ....................325
4.3.3.14 LL/CON/INI/BV-15-C [Connecting to Directed Advertising, Channel Selection Algorithm #1] ..........328
4.3.3.15 LL/CON/INI/BV-16-C [Connection Initiation, Channel Selection Algorithm #2] ....................331
4.3.3.16 LL/CON/INI/BV-17-C [Connecting to Directed Advertising, Channel Selection Algorithm #2] ..........334
4.3.3.17 Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address ...........................................................................................................................................337
4.3.3.18 LL/CON/INI/BV-19-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Ignore Identity Address] .................................................................338
4.3.3.19 LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address] ........................................................................................340
4.3.3.20 LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Accept Identity Address] .................................................................340
4.3.3.21 LL/CON/INI/BV-22-C [Connection Initiation Invalid CRC] ......................................................344
4.3.3.22 LL/CON/INI/BV-23-C [Slave Packets Invalid CRC] .................................................................345
4.3.3.23 LL/CON/INI/BV-24-C [Connection Initiation, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2] .................................................................347
4.3.3.24 LL/CON/INI/BV-25-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled] ........................................................349
4.3.4 SLA .............................................................................351
4.3.4.1 LL/CON/SLA/BV-02-C [Slave Asymmetric Connections] .........................................................351
4.3.4.2 LL/CON/SLA/BV-04-C [Slave Sending Data] .................................................................353
4.3.4.3 LL/CON/SLA/BV-05-C [Slave Receiving Data] .................................................................355
4.3.4.4 LL/CON/SLA/BV-06-C [Slave Sending and Receiving Data] ..................................................357
| 4.3.4.5 | LL/CON/SLA/BV-10-C [Accepting Parameter Update] | 358 |
| 4.3.4.6 | LL/CON/SLA/BV-11-C [Slave Sending Termination] | 361 |
| 4.3.4.7 | LL/CON/SLA/BV-12-C [Slave Accepting Termination] | 363 |
| 4.3.4.8 | LL/CON/SLA/BV-13-C [Slave Supervision Timer] | 364 |
| 4.3.4.9 | LL/CON/SLA/BV-14-C [Feature Setup Response] | 365 |
| 4.3.4.10 | LL/CON/SLA/BV-15-C [Slave Retransmission Request] | 366 |
| 4.3.4.11 | LL/CON/SLA/BV-16-C [Slave Retransmission] | 367 |
| 4.3.4.12 | LL/CON/SLA/BV-17-C [Slave Acknowledgement Repetition] | 368 |
| 4.3.4.13 | LL/CON/SLA/BV-18-C [Slave Lost Negative Acknowledgement] | 369 |
| 4.3.4.14 | LL/CON/SLA/BV-19-C [Slave Request Version] | 371 |
| 4.3.4.15 | LL/CON/SLA/BV-20-C [Slave Respond Version] | 372 |
| 4.3.4.16 | LL/CON/SLA/BV-21-C [Slave Acknowledgement Scheme] | 373 |
| 4.3.4.17 | LL/CON/SLA/BV-22-C [Initiate Feature Exchange] | 374 |
| 4.3.4.18 | LL/CON/SLA/BV-23-C [Initiate Feature Exchange – Master does not support] | 375 |
| 4.3.4.19 | LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept] | 376 |
| 4.3.4.20 | LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject] | 380 |
| 4.3.4.21 | LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision] | 381 |
| 4.3.4.22 | LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update] | 383 |
| 4.3.4.23 | LL/CON/SLA/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption] | 385 |
| 4.3.4.24 | LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity] | 387 |
| 4.3.4.25 | LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only] | 390 |
| 4.3.4.26 | LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity] | 392 |
| 4.3.4.27 | LL/CON/SLA/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points] | 394 |
| 4.3.4.28 | LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked] | 396 |
| 4.3.4.29 | LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects] | 397 |
| 4.3.4.30 | LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure] | 398 |
| 4.3.4.31 | LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure] | 403 |
| 4.3.4.32 | LL/CON/SLA/BV-43-C [Responding to PHY Update Procedure – Symmetric Only] | 409 |
| 4.3.4.33 | LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure] | 409 |
| 4.3.4.34 | LL/CON/SLA/BV-45-C [Protocol Timeout for PHY Update Procedure] | 411 |
| 4.3.4.35 | LL/CON/SLA/BV-46-C [Handling Protocol Collision – Different Procedure – Channel Map] | 413 |
| 4.3.4.36 | LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters] | 414 |
| 4.3.4.37 | LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update] | 416 |
| 4.3.4.38 | LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions] | 418 |
| 4.3.4.39 | LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions] | 422 |
| 4.3.4.40 | LL/CON/SLA/BV-51-C [Protocol Timeout for PHY Update Procedure – No Update Request] | 426 |
| 4.3.4.41 | LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change] | 427 |
| 4.3.4.42 | LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change] | 431 |
| 4.3.4.43 | LL/CON/SLA/BV-54-C [Slave Receiving Data, LE Coded, CI Change] | 435 |
4.3.4.44 LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]..............................................................................................................................439
4.3.4.45 LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded]..............................................................................................................................443
4.3.4.46 LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded]..............................................................................................................................447
4.3.4.47 LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]..............................................................................................................................451
4.3.4.48 LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]..............................................................................................................................454
4.3.4.49 LL/CON/SLA/BV-77-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]..............................................................................................................................457
4.3.4.50 LL/CON/SLA/BV-78-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]..............................................................................................................................457
4.3.4.51 LL/CON/SLA/BV-79-C [Slave Data Length Update – Master does not support; LE 1M PHY]..............................................................................................................................457
4.3.4.52 LL/CON/SLA/BV-80-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]..............................................................................................................................457
4.3.4.53 LL/CON/SLA/BV-81-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]..............................................................................................................................458
4.3.4.54 LL/CON/SLA/BV-82-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY]..............................................................................................................................458
4.3.4.55 LL/CON/SLA/BV-83-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY]..............................................................................................................................458
4.3.4.56 LL/CON/SLA/BV-84-C [Slave Data Length Update – Master does not support; LE Coded PHY]..............................................................................................................................458
4.3.4.57 LL/CON/SLA/BV-85-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]..............................................................................................................................458
4.3.4.58 LL/CON/SLA/BV-86-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]..............................................................................................................................460
4.3.4.59 LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]..................................................................................................................................................................................461
4.3.4.60 LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]..................................................................................................................................................................................463
4.3.4.61 LL/CON/SLA/BI-04-C [Rejecting Connection Change]..................................................................................................................................................................................464
4.3.4.62 LL/CON/SLA/BI-05-C [Slave Connection Control Timer]..................................................................................................................................................................................466
4.3.4.63 LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]..................................................................................................................................................................................468
4.3.4.64 LL/CON/SLA/BI-08-C [Accepting Connection Parameter Request – Illegal Parameters]..................................................................................................................................................................................469
4.3.4.65 LL/CON/SLA/BI-09-C [Responding to PHY Update Procedure – Instant In Past]..................................................................................................................................................................................470
4.3.5 MAS ..................................................................................................................................................................................................................................................472
4.3.5.1 LL/CON/MAS/BV-02-C [Events with Slave Latency]..................................................................................................................................................................................472
4.3.5.2 LL/CON/MAS/BV-03-C [Master Sending Data]..................................................................................................................................................................................474
4.3.5.3 LL/CON/MAS/BV-04-C [Master Receiving Data]..................................................................................................................................................................................476
4.3.5.4 LL/CON/MAS/BV-05-C [Master Sending and Receiving Data]..................................................................................................................................................................................478
4.3.5.5 LL/CON/MAS/BV-07-C [Requesting Parameter Update]..................................................................................................................................................................................479
4.3.5.6 LL/CON/MAS/BV-08-C [Master Sending Termination]..................................................................................................................................................................................481
4.3.5.7 LL/CON/MAS/BV-09-C [Master Accepting Termination]..................................................................................................................................................................................482
4.3.5.8 LL/CON/MAS/BV-10-C [Master Supervision Timer]..................................................................................................................................................................................483
4.3.5.9 LL/CON/MAS/BV-13-C [Feature Setup Request]..................................................................................................................................................................................484
4.3.5.10 LL/CON/MAS/BV-14-C [Master Retransmission Request]..................................................................................................................................................................................485
4.3.5.11 LL/CON/MAS/BV-15-C [Master Retransmission]..................................................................................................................................................................................487
4.3.5.12 LL/CON/MAS/BV-16-C [Master Acknowledgement Repetition]..................................................................................................................................................................................488
4.3.5.13 LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]..................................................................................................................................................................................489
4.3.5.14 LL/CON/MAS/BV-18-C [Master Latency Retransmission Request]..................................................................................................................................................................................491
4.3.5.15 LL/CON/MAS/BV-19-C [Connection Control Timeout]..................................................................................................................................................................................493
4.3.5.16 LL/CON/MAS/BV-20-C [Master Request Version].......................................................... 494
4.3.5.17 LL/CON/MAS/BV-21-C [Master Respond Version]....................................................... 495
4.3.5.18 LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]..................................... 496
4.3.5.19 LL/CON/MAS/BV-23-C [Responding to Feature Exchange]......................................... 497
4.3.5.20 LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]..................... 498
4.3.5.21 LL/CON/MAS/BV-25-C [Initiating Connection Parameter Request – Reject].................... 501
4.3.5.22 LL/CON/MAS/BV-26-C [Initiating Connection Parameter Request – same procedure collision].................................................................................................................. 503
4.3.5.23 LL/CON/MAS/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]................................................................. 505
4.3.5.24 LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]........................................................................................................ 507
4.3.5.25 LL/CON/MAS/BV-29-C [Initiating Connection Parameter Request – remote legacy host]......................................................................................................................... 509
4.3.5.26 LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred_Periodicity].............................................................................................................. 511
4.3.5.27 LL/CON/MAS/BV-31-C [Accepting Connection Parameter Request – preferred anchor points only]............................................................................................................. 514
4.3.5.28 [Accepting Connection Parameter Request]........................................................................ 516
4.3.5.29 LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked].......................................................................................................................... 518
4.3.5.30 LL/CON/MAS/BV-35-C [Accepting Connection Parameter Request – Host rejects]........... 519
4.3.5.31 LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure].............................................. 520
4.3.5.32 LL/CON/MAS/BV-42-C [Initiating PHY Update Procedure – Symmetric Only].................. 524
4.3.5.33 LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure].................................... 525
4.3.5.34 LL/CON/MAS/BV-44-C [Responding to PHY Update Procedure – Symmetric Only]........... 530
4.3.5.35 LL/CON/MAS/BV-45-C [Handling Protocol Collision – Same Procedure].......................... 530
4.3.5.36 LL/CON/MAS/BV-46-C [Protocol Timeout for PHY Update Procedure]............................ 532
4.3.5.37 LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]..................................................................................................................... 534
4.3.5.38 LL/CON/MAS/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Parameters]............................................................................................................ 535
4.3.5.39 LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]........ 537
4.3.5.40 LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions].............................................................................................................................. 541
4.3.5.41 LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]................... 545
4.3.5.42 LL/CON/MAS/BV-52-C [Master Receiving Data, LE Coded, CI Change]............................ 547
4.3.5.43 LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]......................................................................................................................... 550
4.3.5.44 LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded].................................................................................................................. 554
4.3.5.45 LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded]............................. 558
4.3.5.46 LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY].................................................................................... 561
4.3.5.47 LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY].......................................................................................... 564
4.3.5.48 LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY]............................................................................................................................ 567
4.3.5.49 LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY].................................................................................... 570
4.3.5.50 LL/CON/MAS/BV-77-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY].......................................................................................... 571
4.3.5.51 LL/CON/MAS/BV-78-C [Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY].................................................................................... 573
4.3.5.52 LL/CON/MAS/BV-79-C [Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY].................................................................................... 574
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.5.53</td>
<td>LL/CON/MAS/BV-80-C [Master Data Length Update – Slave does not support; LE Coded PHY]</td>
</tr>
<tr>
<td>4.3.5.54</td>
<td>LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]</td>
</tr>
<tr>
<td>4.3.5.55</td>
<td>LL/CON/MAS/BV-82-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]</td>
</tr>
<tr>
<td>4.3.5.56</td>
<td>LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]</td>
</tr>
<tr>
<td>4.3.5.57</td>
<td>LL/CON/MAS/BI-02-C [Master T_Terminate Timer]</td>
</tr>
<tr>
<td>4.3.5.58</td>
<td>LL/CON/MAS/BI-04-C [Master Connection Control Timer]</td>
</tr>
<tr>
<td>4.3.5.59</td>
<td>LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]</td>
</tr>
<tr>
<td>4.3.5.60</td>
<td>LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]</td>
</tr>
<tr>
<td>4.3.6</td>
<td>Both Connected Roles</td>
</tr>
<tr>
<td>4.3.6.1</td>
<td>Constant Tone Extension Request Procedure, IUT Responding, Unsupported</td>
</tr>
<tr>
<td>4.3.6.2</td>
<td>[Periodic Advertising Sync Transfer Procedure, Advertising IUT Initiated]</td>
</tr>
<tr>
<td>4.3.6.3</td>
<td>[Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]</td>
</tr>
<tr>
<td>4.3.6.4</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYS]</td>
</tr>
<tr>
<td>4.3.6.5</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Skipping Events]</td>
</tr>
<tr>
<td>4.3.6.6</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]</td>
</tr>
<tr>
<td>4.3.6.7</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Extreme Timings]</td>
</tr>
<tr>
<td>4.3.6.8</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure]</td>
</tr>
<tr>
<td>4.3.6.9</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses]</td>
</tr>
<tr>
<td>4.3.6.10</td>
<td>[Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]</td>
</tr>
<tr>
<td>4.3.6.11</td>
<td>[Acknowledging Long Control PDUs]</td>
</tr>
<tr>
<td>4.3.6.12</td>
<td>[Rejecting Request to Send Long Control PDUs before Feature Exchange]</td>
</tr>
<tr>
<td>4.3.6.13</td>
<td>[Rejecting Request to Send Long Control PDUs after Feature Exchange]</td>
</tr>
<tr>
<td>4.3.6.14</td>
<td>[Sending Long Control PDUs after Feature Exchange]</td>
</tr>
<tr>
<td>4.3.6.15</td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Unsupported]</td>
</tr>
<tr>
<td>4.3.6.16</td>
<td>Unrequested Constant Tone Extension, IUT Receiving, AoD</td>
</tr>
<tr>
<td>4.3.6.17</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoA</td>
</tr>
<tr>
<td>4.3.6.18</td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Periodic]</td>
</tr>
<tr>
<td>4.3.6.19</td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]</td>
</tr>
<tr>
<td>4.3.6.20</td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Timeout]</td>
</tr>
<tr>
<td>4.3.6.21</td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoA]</td>
</tr>
<tr>
<td>4.3.6.22</td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]</td>
</tr>
<tr>
<td>4.3.6.23</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoD</td>
</tr>
<tr>
<td>4.3.6.24</td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoD]</td>
</tr>
<tr>
<td>4.3.6.25</td>
<td>Unrequested Constant Tone Extension, IUT Receiving, AoA</td>
</tr>
<tr>
<td>4.3.6.26</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection</td>
</tr>
<tr>
<td>4.3.6.27</td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoA, Encrypted Connection]</td>
</tr>
<tr>
<td>4.3.6.28</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection</td>
</tr>
<tr>
<td>4.3.6.29</td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]</td>
</tr>
<tr>
<td>4.3.6.30</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC</td>
</tr>
<tr>
<td>4.3.6.31</td>
<td>Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC</td>
</tr>
<tr>
<td>4.3.6.32</td>
<td>Data Length Update – Preserve Parameters After a PHY Change</td>
</tr>
<tr>
<td>4.3.6.33</td>
<td>Data Length Update – Retransmission During an Update</td>
</tr>
<tr>
<td>4.3.6.34</td>
<td>Data Length Update – Handling Invalid Data Length Responses</td>
</tr>
</tbody>
</table>
4.3.6.35 Data Length Update – Peer Does Not Support LE Coded PHY
4.3.6.36 Constant Tone Extension Disabled on PHY Change
4.4 TIM
4.4.1 ADV
4.4.1.1 LL/TIM/ADV/BV-01-C [Earliest Transmission to Advertiser]
4.4.1.2 LL/TIM/ADV/BV-02-C [Latest Transmission to Advertiser]
4.4.1.3 Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser
4.4.1.4 Extended Advertising, Secondary Channel, Latest Transmission to Advertiser
4.4.2 SLA
4.4.2.1 LL/TIM/SLA/BV-01-C [Adjusting Anchor Point]
4.4.2.2 LL/TIM/SLA/BV-02-C [Earliest Transmission Start to Slave]
4.4.2.3 LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]
4.4.2.4 LL/TIM/SLA/BV-04-C [Shortest Connection Interval]
4.4.2.5 LL/TIM/SLA/BV-05-C [Longest Connection Interval]
4.4.2.6 LL/TIM/SLA/BV-06-C [Earliest Transmissions to Slave]
4.4.2.7 LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]
4.4.2.8 LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]
4.4.2.9 LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update]
4.4.2.10 LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy]
4.4.3 MAS
4.4.3.1 LL/TIM/MAS/BV-01-C [Earliest Transmissions to Master]
4.4.3.2 LL/TIM/MAS/BV-02-C [Latest Transmissions to Master]
4.4.3.3 LL/TIM/MAS/BV-03-C [Initiate Sleep Clock Accuracy Update]
4.4.3.4 LL/TIM/MAS/BV-04-C [Response to Sleep Clock Accuracy Update]
4.4.3.5 LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy]
4.4.3.6 LL/TIM/MAS/BV-06-C [Change Accuracy after Update]
4.4.3.7 LL/TIM/MAS/BV-07-C [Change Accuracy before Update]
4.4.4 SCN
4.4.4.1 Extended Scanning, Secondary Channel, Earliest Transmission to Scanner
4.4.4.2 Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY
4.4.4.3 Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY
4.5 ENC
4.5.1 ADV
4.5.1.1 LL/ENC/ADV/BI-01-C [Scan Request Invalid Address]
4.5.1.2 LL/ENC/ADV/BI-02-C [Master Packets Invalid Address]
4.5.2 SCN
4.5.2.1 LL/ENC/SCN/BI-01-C [Passive Scanning Invalid Address]
4.5.2.2 LL/ENC/SCN/BI-02-C [Active Scanning Invalid Address]
4.5.3 INI
4.5.3.1 LL/ENC/INI/BI-01-C [Slave Packets Invalid Address]
4.5.4 SLA
4.5.4.1 LL/ENC/SLA/BI-01-C [Packets to another Slave]
4.5.5 MAS
4.5.5.1 LL/ENC/MAS/BI-01-C [Packets to another Master]
4.6 FRH
4.6.1 Common PDU Contents
4.6.2 ADV
4.6.2.1 LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]
4.6.3.1  LL/FRH/SLA/BV-01-C [Accepting Channel Map Update] ..................................................... 717
4.6.3.2  LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2] .... 719
4.6.4   MAS ................................................................................................................................. 720
4.6.4.1  LL/FRH/MAS/BV-01-C [Requesting Channel Map Update] ............................................. 720
4.6.4.2  LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2] . 722
4.6.4.3  LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels] ......................... 724
4.7    PAC.................................................................................................................................... 725
4.7.1   Common PDU Contents ..................................................................................................... 725
4.7.2   SLA ..................................................................................................................................... 725
4.7.2.1  LL/PAC/SLA/BV-01-C [Unknown Packet from Master] ..................................................... 725
4.7.3   MAS ..................................................................................................................................... 726
4.7.3.1  LL/PAC/MAS/BV-01-C [Unknown Packet from Slave] ....................................................... 726
4.7.4   Both Connected Roles ........................................................................................................ 727
4.7.4.1  [Control PDUs with Invalid Length] .................................................................................. 727
4.8    SEC....................................................................................................................................... 730
4.8.1   Common PDU Contents ..................................................................................................... 730
4.8.2   ADV .................................................................................................................................... 731
4.8.2.1  LL/SEC/ADV/BV-01-C [Advertising With Static Address] ................................................. 731
4.8.2.2  LL/SEC/ADV/BV-02-C [Privacy - Non-connectable Undirected Advertising, non- resolvable private address] ......................................................................................... 733
4.8.2.3  LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address] ........................................................................................................... 735
4.8.2.4  LL/SEC/ADV/BV-04-C [Network Privacy - Scannable Advertising, non-resolvable private address] ...................................................................................................................... 737
4.8.2.5  LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address] ............................................................................................................................... 739
4.8.2.6  LL/SEC/ADV/BV-06-C [Network Privacy - Undirected Connectable Advertising no Local IRK, no peer IRK] ........................................................................................................ 741
4.8.2.7  LL/SEC/ADV/BV-07-C [Network Privacy - Undirected Connectable Advertising with Local IRK, no peer IRK] ........................................................................................................ 743
4.8.2.8  LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK] ...................................................................................................... 745
4.8.2.9  LL/SEC/ADV/BV-09-C [Network Privacy - Undirected Connectable Advertising without Local IRK and with peer IRK] ..................................................................................... 747
4.8.2.12 LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK] .......................................................................................... 753
4.8.2.16 LL/SEC/ADV/BV-16-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Ignore Identity Address] .................................................................... 761
4.8.2.17 LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address] ........................................................................ 763
4.8.2.18 LL/SEC/ADV/BV-18-C [Device Privacy - Scannable Advertising, resolvable private address, Accept Identity Address] ............................................................................................... 765
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8.20</td>
<td>LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address]</td>
</tr>
<tr>
<td>4.8.3</td>
<td>SCN ...........................................................................................................</td>
</tr>
<tr>
<td>4.8.3.1</td>
<td>LL/SEC/SCN/BV-01-C [Random Address Scanning] .......................................</td>
</tr>
<tr>
<td>4.8.4</td>
<td>INI ...........................................................................................................</td>
</tr>
<tr>
<td>4.8.5</td>
<td>SLA ...........................................................................................................</td>
</tr>
<tr>
<td>4.8.5.1</td>
<td>LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup] ...................................</td>
</tr>
<tr>
<td>4.8.5.2</td>
<td>LL/SEC/SLA/BV-02-C [Slave Pause Encryption] .........................................</td>
</tr>
<tr>
<td>4.8.5.3</td>
<td>LL/SEC/SLA/BV-03-C [Slave Pause Encryption Sending Data] .........................</td>
</tr>
<tr>
<td>4.8.5.4</td>
<td>LL/SEC/SLA/BV-04-C [Slave Sending LL_REJECT_IND] ....................................</td>
</tr>
<tr>
<td>4.8.5.5</td>
<td>LL/SEC/SLA/BV-05-C [Slave Receiving Encrypted Data] ................................</td>
</tr>
<tr>
<td>4.8.5.6</td>
<td>LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled] ......</td>
</tr>
<tr>
<td>4.8.5.7</td>
<td>LL/SEC/SLA/BV-07-C [Responding to LL_PING_REQ] ....................................</td>
</tr>
<tr>
<td>4.8.5.8</td>
<td>LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ] ..................................</td>
</tr>
<tr>
<td>4.8.5.9</td>
<td>LL/SEC/SLA/BV-09-C [Modified Authentication Payload Timeout] .....................</td>
</tr>
<tr>
<td>4.8.5.10</td>
<td>LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]</td>
</tr>
<tr>
<td>4.8.5.11</td>
<td>LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND] ...............................</td>
</tr>
<tr>
<td>4.8.5.13</td>
<td>LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC] ...............................</td>
</tr>
<tr>
<td>4.8.5.14</td>
<td>LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header] .........................</td>
</tr>
<tr>
<td>4.8.5.15</td>
<td>LL/SEC/SLA/BI-05-C [Slave Receiving unexpected PDU during encryption start] .....</td>
</tr>
<tr>
<td>4.8.6</td>
<td>MAS ...........................................................................................................</td>
</tr>
<tr>
<td>4.8.6.1</td>
<td>LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup] ..................................</td>
</tr>
<tr>
<td>4.8.6.2</td>
<td>LL/SEC/MAS/BV-02-C [Master Pause Encryption] .........................................</td>
</tr>
<tr>
<td>4.8.6.3</td>
<td>LL/SEC/MAS/BV-03-C [Master Receiving LL_REJECT_IND] ...............................</td>
</tr>
<tr>
<td>4.8.6.4</td>
<td>LL/SEC/MAS/BV-04-C [Master Encryption: Sending Data before LL_ENC_RSP] ........</td>
</tr>
<tr>
<td>4.8.6.5</td>
<td>LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]</td>
</tr>
<tr>
<td>4.8.6.6</td>
<td>LL/SEC/MAS/BV-06-C [Initiate LE Ping procedure when encryption is enabled] ......</td>
</tr>
<tr>
<td>4.8.6.7</td>
<td>LL/SEC/MAS/BV-07-C [Responding to LL_PING_REQ] ....................................</td>
</tr>
<tr>
<td>4.8.6.8</td>
<td>LL/SEC/MAS/BV-08-C [No response to LL_PING_REQ] ..................................</td>
</tr>
<tr>
<td>4.8.6.9</td>
<td>LL/SEC/MAS/BV-09-C [Modified Authentication Payload Timeout] .....................</td>
</tr>
<tr>
<td>4.8.6.10</td>
<td>LL/SEC/MAS/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]</td>
</tr>
<tr>
<td>4.8.6.14</td>
<td>LL/SEC/MAS/BV-14-C [Master Receiving unexpected PDU during encryption start] ...</td>
</tr>
<tr>
<td>4.8.6.18</td>
<td>LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC] ...............................</td>
</tr>
</tbody>
</table>
4.10 CIS

4.10.1 Common Variables, Timing, and Procedures ............................................. 834

4.10.2 MAS .............................................................................................................. 837

4.10.3 SLA .............................................................................................................. 869

4.10.4 LL/SEC/MAS/BL-07-C [Master Pause Encryption Sending Data] ........................... 826

4.10.5 LL/SEC/MAS/BL-08-C [Master Encryption: Sending Data and Not Response] .............. 827

4.10.6 LL/SEC/MAS/BL-09-C [Master Encryption: Sending Data and Not Request] .................. 828

4.9 Data Flow ........................................................................................................... 830

4.9.1 Both Connected Roles .................................................................................. 830

4.9.1.1 [Transmit Fragmented L2CAP Header] ...................................................... 830

4.9.1.2 [Receive Fragmented L2CAP Header] ...................................................... 832

4.10.1.1 Acknowledgement Sequence ................................................................... 834

4.10.1.2 Timing Requirements ............................................................................ 835

4.10.1.3 Default Values for Set CIG Parameters Commands .............................. 835

4.10.1.4 Configuring the ISO Data Path for HCI .................................................. 837

4.10.1.5 Identification of CISes ............................................................................ 837

4.10.2.1 [CIS Setup Procedure, Master Initiated] ............................................... 837

4.10.2.2 LL/CIS/MAS/BV-03-C [CIS Setup Procedure, Master Initiated, Unsupported] ............. 840

4.10.2.3 LL/CIS/MAS/BV-04-C [New Channel Map] ......................................... 842

4.10.2.4 LL/CIS/MAS/BV-10-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Interleaved CIG] .................................. 843

4.10.2.5 LL/CIS/MAS/BV-11-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Sequential] .................................. 845

4.10.2.6 LL/CIS/MAS/BV-20-C [Set Encryption After CIS Established] .................. 848

4.10.2.7 Connected Isochronous Stream Using Non-Test Command, Master Initiated .......... 849

4.10.2.8 LL/CIS/MAS/BV-30-C [Isochronous Channels Host Support Feature Bit] ............. 851

4.10.2.9 LL/CIS/MAS/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG, Master] .......................... 853

4.10.2.10 LL/CIS/MAS/BV-09-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential, Master] .................................. 857

4.10.2.11 [Acknowledgement Scheme, Master] .................................................. 859

4.10.2.12 LL/CIS/MAS/BV-17-C [Flushing of Packets in CIS, Master] ......................... 861

4.10.2.13 LL/CIS/MAS/BV-18-C [Bursting of Payloads in CIS, Master] ......................... 863


4.10.3.1 [CIS Setup Response Procedure, Slave] .................................................. 869

4.10.3.2 LL/CIS/SLA/BV-02-C [CIS Setup Response Procedure, Slave, Reject Response] .... 872

4.10.3.3 LL/CIS/SLA/BV-03-C [CIS Map Update] .............................................. 873

4.10.3.4 LL/CIS/SLA/BV-22-C [CIS Request Event Not Set] .................................. 874

4.10.3.5 LL/CIS/SLA/BV-07-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG, Slave] .......................... 876

4.10.3.6 LL/CIS/SLA/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential, Slave] .................................. 879

4.10.3.7 [Acknowledgement Scheme, Slave] ...................................................... 882

4.10.3.8 LL/CIS/SLA/BV-14-C [Flushing of Packets in CIS, Slave] ............................. 884

4.10.3.9 LL/CIS/SLA/BV-15-C [Bursting of Payloads in CIS, Slave] ............................ 886

4.10.3.10 LL/CIS/SLA/BV-16-C [Deterministic Packet Transmission in CIS, Slave] .......... 889
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.10.4</td>
<td>Both Connected Roles</td>
</tr>
<tr>
<td>4.10.4.1</td>
<td>[Sending data in Unidirectional CIS]</td>
</tr>
<tr>
<td>4.10.4.2</td>
<td>[Receiving data in Unidirectional CIS]</td>
</tr>
<tr>
<td>4.10.4.3</td>
<td>[Sending and Receiving Data in Bidirectional CIS]</td>
</tr>
<tr>
<td>4.10.4.4</td>
<td>[Retransmissions]</td>
</tr>
<tr>
<td>4.10.4.5</td>
<td>[CIS Terminate Procedure, Initiated]</td>
</tr>
<tr>
<td>4.10.4.6</td>
<td>[CIS Terminate Procedure, Accepting]</td>
</tr>
<tr>
<td>4.10.4.7</td>
<td>[CIS Updating Peer Clock Accuracy]</td>
</tr>
<tr>
<td>4.10.4.8</td>
<td>[Error Updating Peer Clock Accuracy]</td>
</tr>
<tr>
<td>4.11</td>
<td>BIS</td>
</tr>
<tr>
<td>4.11.1</td>
<td>Common Parameters</td>
</tr>
<tr>
<td>4.11.1.1</td>
<td>Configuring the ISO Data Path for HCI</td>
</tr>
<tr>
<td>4.11.1.2</td>
<td>Timing Requirements</td>
</tr>
<tr>
<td>4.11.2</td>
<td>Common PDU Contents</td>
</tr>
<tr>
<td>4.11.3</td>
<td>BRD</td>
</tr>
<tr>
<td>4.11.3.1</td>
<td>[Broadcast Isochronous Stream Setup]</td>
</tr>
<tr>
<td>4.11.3.2</td>
<td>LL/BIS/BRD/BV-04-C [Data Transmission in Multiple Broadcast Isochronous Streams, Interleaved Big]</td>
</tr>
<tr>
<td>4.11.3.3</td>
<td>LL/BIS/BRD/BV-05-C [Data Transmission in Multiple Broadcast Isochronous Streams, Sequential]</td>
</tr>
<tr>
<td>4.11.3.4</td>
<td>LL/BIS/BRD/BV-07-C [Bursting of Packets in Broadcast Isochronous Stream]</td>
</tr>
<tr>
<td>4.11.3.5</td>
<td>LL/BIS/BRD/BV-08-C [Pre-transmissions in Broadcast Isochronous Stream]</td>
</tr>
<tr>
<td>4.11.3.6</td>
<td>LL/BIS/BRD/BV-09-C [Broadcast Isochronous Group Channel Map Update Procedure]</td>
</tr>
<tr>
<td>4.11.3.7</td>
<td>LL/BIS/BRD/BV-10-C [Isochronous Broadcaster Terminates BIS Stream]</td>
</tr>
<tr>
<td>4.11.3.8</td>
<td>Broadcast Isochronous Stream Using Non-Test Command</td>
</tr>
<tr>
<td>4.11.3.9</td>
<td>LL/BIS/BRD/BV-16-C [Maximum Supported Broadcast Isochronous Groups]</td>
</tr>
<tr>
<td>4.11.4</td>
<td>SNC</td>
</tr>
<tr>
<td>4.11.4.1</td>
<td>Broadcast Isochronous Stream Synchronization Setup</td>
</tr>
<tr>
<td>4.11.4.2</td>
<td>LL/BIS/SNC/BV-04-C [Data Reception in Multiple Broadcast Isochronous Streams]</td>
</tr>
<tr>
<td>4.11.4.3</td>
<td>LL/BIS/SNC/BV-07-C [Bursting of Packets in Broadcast Isochronous Stream]</td>
</tr>
<tr>
<td>4.11.4.4</td>
<td>LL/BIS/SNC/BV-08-C [Pre-transmissions in Broadcast Isochronous Stream]</td>
</tr>
<tr>
<td>4.11.4.5</td>
<td>LL/BIS/SNC/BV-09-C [Broadcast Isochronous Group Channel Map Update Procedure]</td>
</tr>
<tr>
<td>4.11.4.6</td>
<td>LL/BIS/SNC/BV-10-C [Broadcast Isochronous Stream Termination]</td>
</tr>
<tr>
<td>4.11.4.7</td>
<td>LL/BIS/SNC/BV-11-C [Loss of Sync with an Isochronous Broadcaster]</td>
</tr>
<tr>
<td>4.11.4.8</td>
<td>LL/BIS/SNC/BV-12-C [Broadcast Isochronous Stream Synchronization, Number of BISes Not Supported]</td>
</tr>
<tr>
<td>4.12</td>
<td>Isochronous Testing</td>
</tr>
<tr>
<td>4.12.1</td>
<td>Common Test Parameter Settings</td>
</tr>
<tr>
<td>4.12.2</td>
<td>[ISO Transmit Test Mode, CIS]</td>
</tr>
<tr>
<td>4.12.2.1</td>
<td>LL/IST/MAS/BV-01-C</td>
</tr>
<tr>
<td>4.12.2.2</td>
<td>LL/IST/SLA/BV-01-C</td>
</tr>
<tr>
<td>4.12.3</td>
<td>LL/IST/BRD/BV-01-C [ISO Transmit Test Mode, BIS]</td>
</tr>
<tr>
<td>4.12.4</td>
<td>[ISO Receive Test Mode, CIS]</td>
</tr>
<tr>
<td>4.12.4.1</td>
<td>LL/IST/MAS/BV-03-C</td>
</tr>
<tr>
<td>4.12.4.2</td>
<td>LL/IST/SLA/BV-03-C</td>
</tr>
<tr>
<td>4.12.5</td>
<td>LL/IST/SNC/BV-01-C [ISO Receive Test Mode, BIS]</td>
</tr>
<tr>
<td>4.12.6</td>
<td>[ISO Receive Test Mode missing PDU]</td>
</tr>
<tr>
<td>4.12.6.1</td>
<td>LL/IST/MAS/BV-05-C</td>
</tr>
<tr>
<td>4.12.6.2</td>
<td>LL/IST/SLA/BV-05-C</td>
</tr>
<tr>
<td>4.13</td>
<td>PCL</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>4.13.1</td>
<td>RF Test Conditions</td>
</tr>
<tr>
<td>4.13.2</td>
<td>Common PDU Contents</td>
</tr>
<tr>
<td>4.13.2.1</td>
<td>Power Control PDUs</td>
</tr>
<tr>
<td>4.13.3</td>
<td>Both Connected Roles</td>
</tr>
<tr>
<td>4.13.3.1</td>
<td>[Path Loss Monitoring]</td>
</tr>
<tr>
<td>4.13.3.2</td>
<td>[Power Control Request – Initiate]</td>
</tr>
<tr>
<td>4.13.3.3</td>
<td>[Power Control Response]</td>
</tr>
<tr>
<td>4.13.3.4</td>
<td>[Power Configuration Response, Min and Max Power Level Reached]</td>
</tr>
<tr>
<td>4.13.3.5</td>
<td>[Set Acceptable Power Reduction Value]</td>
</tr>
<tr>
<td>4.13.3.6</td>
<td>[Properly handle a Power Request PDU when waiting for a Power Response PDU]</td>
</tr>
<tr>
<td>4.13.3.7</td>
<td>[Power Control Request using an unsupported PHY]</td>
</tr>
<tr>
<td>4.13.3.8</td>
<td>[Power Change Indication on PHY Change]</td>
</tr>
<tr>
<td>4.13.3.9</td>
<td>[Power Control Request when a CIS has been established]</td>
</tr>
<tr>
<td>4.13.3.10</td>
<td>[Power Control Response when a CIS has been established]</td>
</tr>
<tr>
<td>4.13.3.11</td>
<td>[Remote Transmit Power Level Request]</td>
</tr>
<tr>
<td>4.13.3.12</td>
<td>[Power Control Response with RF Path Compensation]</td>
</tr>
<tr>
<td>4.13.3.13</td>
<td>[Max and Min Power Level Response at Max and Min Power]</td>
</tr>
</tbody>
</table>

5 Test Case Mapping

6 Revision History and Contributors
1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and Test Cases (TC) to test the Bluetooth Low Energy Link Layer (LL).

The objective of this test suite is to provide a basis for interoperability for Bluetooth devices giving a high probability of air interface interoperability between different manufacturers' Bluetooth devices.
2 References, Definitions, and Abbreviations

2.1 References

This Bluetooth document incorporates, by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For the purpose of this Bluetooth document, the definitions and abbreviations found in [1], [2], and [3] apply.

[1] Specification of the Bluetooth System, Versions 4.0 or later
[7] Bluetooth Test Suite for RF-PHY
[9] Bluetooth Core IXIT proforma
[15] Specification of the Bluetooth System, Volume 2, Part E (versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher) (Host Controller Interface Functional Specification), Version 5.2 or later
[16] Specification of the Bluetooth System, Volume 6, Part D (Message Sequence Charts), Version 5.2 or later
[18] Supplement to Bluetooth Core Specification, Part A, Version 7 or later
[19] Supplement to Bluetooth Core Specification, Part A, Version 5.2 or later
3 Test Suite Structure (TSS)

3.1 Test Strategy

The objective of testing is to ensure interoperability and functionality of the [2] implementations by conformance tests.

Conformance tests will be realized in test equipment providing a testing configuration with a Lower and Upper Tester ([6], General Concepts, p. 27).

The intent is to test mandatory and optional LL requirements and their combinations, in the protocol specification as applicable by the ICS proforma in [4] and [6]. Static conformance requirements such as device addressing and correct packet formats are verified with the tests for the dynamic requirements.

The bullet points in Figure 3.1 outline the test functionality grouped according to protocol procedures, orthogonal functions and testing objectives. Features and roles subject to testing appear in the groups as subgroups or functions.
- Device Discovery
  - Advertising
    - Non connectable Events
    - Connectable Events
    - Device Filtering
    - Advertising Data
    - Invalid Request Packets
    - Continuous Advertising
    - Combining Roles and Advertising
  - Scanning
    - Passive
    - Active
    - Device Filtering
    - Invalid Advertising Packets
    - Continuous Scanning
    - Combining Roles and Scanning
- Connection Handling
  - Advertising
    - Accepting Connections
    - Invalid Packets from Initiators
  - Initiating
    - Initiating
    - Invalid Packets from Advertisers and Slave
    - Addressing
  - Slave
    - Acknowledgement Scheme
    - Event Handling
    - Addressing
    - Invalid Packets from Master
    - Data Transfer
    - Accepting Parameter Update
    - Termination
    - Combining Slave Role
    - Initiate and Accept Parameter Request
  - Master
    - Acknowledgement Scheme
    - Addressing
    - Invalid Packets from Slave
    - Data Transfer
    - Parameter Update
    - Termination
    - Combining Master Role
    - Initiate and Accept Parameter Request
- Timing in Active and Low Power Modes
- Radio Frame Encoding
  - Incorrect Frames
- Frequency Hopping
- Packet Formats
- Security
  - Private Addressing
  - Encryption Mode Change
- Isochronous Streams
  - Connected Isochronous Stream
  - Broadcast Isochronous Stream
  - Isochronous Test Mode

Figure 3.1: Test Suite Structure

The test grouping reflects the protocol procedures (behavior for device discovery, connection handling, and security) as well as testing for specific aspects (timing, formats, frequency hopping) in any of the protocol procedures.

Testing and qualification of the Angle of Arrival / Angle of Departure functionality is limited to the packet format, control procedures, Constant Tone Extension presence, and that the Controller generates IQ samples and delivers them to the Host. Verification of the validity of the IQ values and antenna performance is not included.
3.2  Test Groups

The groups for the protocol procedures are for testing requirements with parameter variation, timing variation without drift, acknowledgements, CRC checking and addressing. As timing, radio frame encoding and packet formats are grouped in the protocol requirements and may apply to all protocol functions they are also separated to their own top level groups in the testing objectives.

3.2.1  Device Discovery

Test the device discovery procedures: Advertising and scanning modes between multiple devices, verifying correct addressing of devices and the correct content and checksums in packets. Test device filtering and data transfer in the procedures. Measure the timing for advertising events and packets.

3.2.2  Connection Handling

Test the connection setup, flow control and acknowledgement schemes and terminating connections. Test the addressing used in the procedures and the correct content of packets. Measure the timing of connection events and packets using timing parameters between the minimum and maximum slave latencies supported.

Test updating the connection parameters with parameters supported by the IUT.

Test data transfer with connections using positive and negative acknowledgements and flow control and checksums for multiple devices. Test different payload lengths.

Test explicit termination and termination by supervision timers for the protocol roles.

3.2.3  Timing

Test procedures during connections, varying timing within allowed clock drift against the IUT, observing connection parameters for the different roles. That timing deviation by the IUT is within clock drift and jitter is verified in measurements done in the tests in the groups ‘DDI’ and ‘CON’.

3.2.4  Radio Frame Encoding

Test rejection of packets with an invalid preamble or access address, i.e. incorrect content in LL packet sections other than the PDU and CRC.

3.2.5  Frequency Hopping

Test the data channel selection algorithm.

3.2.6  Packet Formats

Test rejection of packets with invalid logical structure for dependencies or content.

3.2.7  Security

Test the encryption mode change operation. Test encrypted address usage during device discovery and encryption of packets in connected operation.

3.2.8  Connected Isochronous Stream

Test one or multiple point-to-point isochronous logical transports. Variable flushing periods for payloads, variable size data contents in packets, and variable number of events that allow a range of isochronous bit rates, latencies, and re-transmissions are tested.
3.2.9 Broadcast Isochronous Stream
Test logical transports that transmit one or more isochronous data streams to in-range scanning devices.

3.2.10 Isochronous Test Mode
Test the Isochronous test mode for testing the Connected or Broadcast Isochronous Stream transmission and reception of isochronous data.

3.2.11 Power Control
Test procedures to control the transmit power of remote device and requesting path loss monitoring.

3.3 Behavior Tests
The tests include both valid and invalid behavior as described by the protocol requirements. In valid behavior tests the IUT is triggered with valid message sequences and is expected to perform the protocol function. In invalid behavior tests the IUT is triggered with sequences containing invalid messages, missing messages, messages of incorrect type or content and is expected to recover and resume the protocol function.

3.3.1 Valid Behavior (BV) Tests
These tests verify that the IUT reacts in conformity with the specification when receiving with valid message sequences.

3.3.2 Invalid Behavior (BI) Tests
These tests verify that the IUT reacts in conformity with the specification when receiving sequences containing invalid messages, missing messages, messages of incorrect type or content and is expected to recover and resume the protocol function.

3.4 Test Realization
An Upper Tester interface is required in the conformance testing configuration. A subset of the HCI [11] is used as the TCI. In test realization, an Implementation under Test (“IUT”) is assumed to implement the HCI commands and events either explicitly or implicitly referenced in a supported procedure requirement in the protocol specification.
4 Test Cases (TC)

4.1 Introduction

4.1.1 Test Case Naming Conventions

Test cases shall be assigned unique identifiers per the conventions in [2]. The convention used here is 
<spec abbreviation>/<IUT role>/<class>/<feat>/<func>/<subfunc>/<cap>/<xx>-<nn>-<y>.

Bolded ID parts shall appear in the order prescribed. Non-bolded ID parts (if applicable) shall appear between the bolded parts. The order of the non-bolded parts may vary from test suite to test suite, but shall be consistent within each individual test suite.

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Spec Abbreviation Identifier &lt;spec abbreviation&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>Link Layer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Feature Identifier &lt;feat&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIS</td>
<td>Broadcast Isochronous Stream</td>
</tr>
<tr>
<td>CIS</td>
<td>Connected Isochronous Stream</td>
</tr>
<tr>
<td>CON</td>
<td>Connection Handling</td>
</tr>
<tr>
<td>DDI</td>
<td>Device Discovery</td>
</tr>
<tr>
<td>ENC</td>
<td>Radio Frame Encoding</td>
</tr>
<tr>
<td>FRH</td>
<td>Frequency Hopping</td>
</tr>
<tr>
<td>IST</td>
<td>Isochronous Testing</td>
</tr>
<tr>
<td>PAC</td>
<td>Packet Formats</td>
</tr>
<tr>
<td>SEC</td>
<td>Security</td>
</tr>
<tr>
<td>TIM</td>
<td>Timing</td>
</tr>
<tr>
<td>PCL</td>
<td>Power Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Function Identifier &lt;func&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADV</td>
<td>Advertiser role</td>
</tr>
<tr>
<td>INI</td>
<td>Initiator role</td>
</tr>
<tr>
<td>MAS</td>
<td>Master role</td>
</tr>
<tr>
<td>SCN</td>
<td>Scanner role</td>
</tr>
<tr>
<td>SLA</td>
<td>Slave role</td>
</tr>
</tbody>
</table>

Table 4.1: TC Feature naming convention for LL
4.1.2 Conformance
When conformance is claimed, all capabilities indicated as mandatory for this Specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth Qualification Program.

The Bluetooth Qualification Program may employ tests to verify implementation robustness. The level of implementation robustness that is verified varies from one Specification to another and may be revised for cause based on interoperability issues found in the market.

Such tests may verify:

- That claimed capabilities may be used in any order and any number of repetitions that is not excluded by the Specification, OR
- That capabilities enabled by the implementations are sustained over durations expected by the use case, OR
- That the implementation gracefully handles any quantity of data expected by the use case, OR
- That in cases where more than one valid interpretation of the Specification exist, the implementation complies with at least one interpretation and gracefully handles other interpretations OR
- That the implementation is immune to attempted security exploits.

A single execution of each of the required tests is required in order to constitute a pass verdict. However, it is noted that in order to provide a foundation for interoperability, it is necessary that a qualified implementation consistently and repeatedly pass any of the applicable tests.

In any case, where a member finds an issue with the Test Plan Generator, the Test Case as described in the Test Suite, or with the Test System utilized, the Member is required to notify the responsible party via an errata request such that the issue may be addressed.

Tests are divided into different groups. For each group of tests common information and the value of the parameters to be used is provided. These parameters can be redefined in each test. The output obtained from the IUT is expected to match with what is defined in the verdict section of each test description.

4.1.3 Common Packet Contents
Contents of radio frames (LL packet sections other than the PDU and CRC) from the IUT are expected to match the descriptions. The bit ordering when defining fields within the packet or Protocol Data Unit (PDU) in the Link Layer specification follows the Little Endian format. The following rules apply:

- The Least Significant Bit (LSB) corresponds to \( b_0 \);
- The LSB is the first bit sent over the air;
- In illustrations, the LSB is shown on the left side.

4.1.3.1 Access Address Used
When the IUT is in the advertising state or slave role, the default value for the data channel access address is 0x456789AB. With the same convention the advertising channel access address (see below) is expressed with a hexadecimal number as 0x8E89BED6.
Where an invalid access address is used for a frame, it is corrupted by inverting a single bit and the test procedure steps will by repetition reflect the probability of the error being reverted. The corruption is done alternating from the last bit before the PDU and from the second bit after the preamble, such that the addresses used are invalid data for the tests but still according to the protocol transfer syntax.

4.1.4 Common Message Contents

Contents of PDU and CRC parts of packets from the IUT are expected to match the descriptions common for the procedures of a group of tests.

Strings are enclosed in quotes and character strings in double quotes.

4.1.4.1 Checksums Used

Where an invalid CRC is used for a PDU, the checksum is corrupted by negating all of the bits in order to have a small probability of the error being reverted in a test.

4.1.4.2 Device Addresses Used

When the IUT is in advertising state or slave role, a default value for the scanning, initiating or master address used is 0x123456789ABC. When the IUT is in scanning state, initiating state or master role, a default value for the address used for the state of advertising or the role of slave is 0x456789ABCDEF.

When the Lower Tester address is required to differ from the IUT address in the most significant octet, this shall be done by XORing the address with 0xA50000_000000. When the Lower Tester address is required to differ from the IUT address in the least significant octet, this shall be done by XORing the address with 0x000000_00005A. When the Lower Tester address is required to differ from the IUT address in both the most and least significant octets, this shall be done by XORing the address with 0x5A0000_0000A5.

4.1.5 Common Initial and Final Conditions

A test procedure description by test steps assumes sets of initial and final conditions of the protocol implementation. In addition, each test procedure may be parameterized for purposes of varying test inputs, or by IUT specific values to indicate e.g. supported ranges to the test.

Common parameters to tests are listed below. Each test may specify additional parameters. ICS and IXIT parameters are identified in [5]. Test procedures use typical parameter values from the protocol specification for illustration. The values supported by the IUT and indicated by IXIT entries, e.g. for timing intervals may be used instead in order to execute a test. Note that the verdict criteria may have to be adjusted in cases where the input parameters are essential for the verdict criteria.

Parameter: All tests include the parameters that indicate the address of the IUT: LL_device_address.

Parameter: For the device discovery tests where a device name is transmitted in LL packets, use “LT” for the Lower Tester and “IUT” for the IUT.

The initial state for an IUT before a test procedure can be executed is stated in each test. This state is typically equivalent to that after the execution of a list of preamble steps. Preamble steps used in multiple tests are defined on group level in order to avoid re-specifying them in each applying test.

If a test does not mention unmasking events, the IUT shall unmask all events mentioned in that test before the test procedure is executed.
If a test covers optional features and does not mention the feature exchange procedure, the IUT shall perform the feature exchange procedure before the test procedure is executed.

It is assumed that the IUT will begin all new connections using the LE 1M PHY in both directions.

**State: Standby**

A precondition in addition to the test specific preamble steps is that it is assumed that the IUT is powered on and active. This is equivalent to being initialized with execution of the following preamble steps:

```plaintext
msc variables LE_supported: LMP_Features, br_edr_not_supported: LMP_Features,
   device_address: BD_ADDR, le_features : LMP_Features, events_supported: Event_Mask

Lower Tester       IUT       Upper Tester

TCI is connected to the IUT HC

Power on

The BER on the RF is less than 0.1%

Idle Preamble Steps

HCI_Reset

HCI_Command_Complete (0x00)

HCI_Read_Local_Supported_Features

HCI_Command_Complete (0x00, le_supported & br_edr_not_supported)

HCI_LE_Read_Local_Supported_Features

HCI_Command_Complete (0x00, le_features)

HCI_LE_Set_Event_Mask (events_supported)

HCI_Command_Complete (0x00)

HCI_LE_Set_Event_Mask (le_events_supported)

HCI_Command_Complete (0x00)

HCI_LE_Set_Event_Mask (events_page2_supported)

HCI_Command_Complete (0x00)
```

*Figure 4.1: Standby Preamble Steps*
The preamble steps do not have to be executed in case a previous test case execution is completed with all postamble steps. Otherwise at least a part of the above steps might have to be executed.

**State: Device Address Set**

```plaintext
State: Device Address Set
Dir: msc variables address_type: Address_Type, device_address: Address

alt

Random Address Generation Preamble Steps Executed

HCI_LE_Set_Random_Address
(random_address)

HCI_Command_Complete
(0x00)

alt

otherwise

opt

LL 2/1

HCI_Read_BD_ADDR

HCI_Command_Complete
(0x00, device_address)

opt

LL 2/2 AND not (LL 2/1)

HCI_LE_Set_Random_Address
(random_address)

HCI_Command_Complete
(0x00)
```

*Figure 4.2: Device Address Set Preamble Steps*

The preamble steps above read or set an address supported by the IUT.
State: Specific White Listed

The HCI commands to set the IUT to apply filtering to specific devices with the parameters of a list of device addresses are displayed in the sequence chart below.

```
msc variables device_addresses: Device_Address[], address_types: Address_Type[], policy_target: Policy_Target, policy: Policy
```

Figure 4.3: Specific White Listed Preamble Steps

Step 2 is optional when the preamble steps are used only to set the filter policy. The preamble steps do not include getting the maximum count of white list entries, since the tests use only one entry at a time.
State: Buffer Size Read

The HCI commands to determine the usage of LE or BR+EDR data buffers for tests transferring data are displayed in the sequence chart below:

**msc variables** br_edr_not_supported: LMP_Features, data_packet_length: Data_Packet_Length, num_data_packets: Num_Data_Packets, Boolean: le_buffers_not_supported

![Sequence Diagram]

**Figure 4.4: Buffer Size Read Preamble Steps**
The outcome of the preamble steps is the information about data primitives and parameters supported by the IUT as well as the configuration of the data primitive parameters supported by the Upper Tester to the IUT.

### 4.1.5.1 ADV

The common initial states of the advertising state.

**State: Advertising Parameters Set**

The HCI commands to set the parameters’ advertising interval and advertising channel map are displayed in the sequence chart below.

```
msc variables adv_interval_min: Adv_Interval, adv_interval_max: Adv_Interval, advertising_type: Advertising_Type, address_type_own: Address_Type, address_type_direct: Address_Type, adv_channel_map: Adv_Channel_Map, advertising_filter_policy: Advertising Filter Policy
```

![Sequence Chart for Advertising Parameters Set Preamble Steps]

*Figure 4.5: Advertising Parameters Set Preamble Steps*
State: Non-Connectable Advertising

The HCI commands to set the IUT to advertise in non-connectable mode with parameters are displayed in the sequence chart below.

**msc variables** adv_interval_min: Adv_Interval, adv_interval_max: Adv_Interval, advertising type: Advertising_Type, address_type_own: Address_Type, address_type_direct: Address_Type, adv_channel_map: Adv_Channel_Map, advertising filter policy: Advertising Filter Policy

**Figure 4.6: Non-Connectable Advertising Preamble Steps**
State: Undirected Advertising

The HCI commands to set the IUT to advertise in connectable undirected mode with parameters are displayed in the sequence chart below.

**mnc variables** adv_interval_min: Adv_Interval, adv_interval_max: Adv_Interval, adv_type: Advertising_Type, address_type_own: Address_Type, address_type_direct:Address_Type, adv_channel_map: Adv_Channel_Map, advertising filter policy: Advertising Filter Policy, scan_response_data_length: Data_Len, scan_response_data: Data

---

**Figure 4.7: Undirected Advertising Preamble Steps**
State: Discoverable Advertising

The HCI commands to set the IUT to advertise in discoverable undirected mode with parameters are displayed in the sequence chart below.

**msc variables**
- `adv_interval_min`: `Adv_Interval`
- `adv_interval_max`: `Adv_Interval`
- `adv_type`: `Advertising_Type`
- `address_type_own`: `Address_Type`
- `address_type_direct`: `Address_Type`
- `adv_channel_map`: `Adv_Channel_Map`
- `adv_filtering_policy`: `Advertising Filter Policy`
- `scan_response_data_length`: `Data_Len`
- `scan_response_data`: `Data`

**Figure 4.8: Discoverable Advertising Preamble Steps**
State: Periodic Advertising

The HCI commands to set the IUT to advertise in periodic advertising mode with parameters as displayed in the sequence chart below.

**Figure 4.9: Periodic Advertising Preamble Steps**

State: Directed Advertising

The HCI commands to set the IUT to advertise using directed advertising with parameters events are displayed in the Figure 4.10.

**Figure 4.10: Directed Advertising Preamble Steps**
State: Extended Advertising

The HCI commands to set the IUT to advertise using extended advertising with parameters as displayed in the sequence chart below.

![Sequence chart for extended advertising](image)

**Figure 4.11: Extended Advertising Preamble Steps**

Extended Advertising msc variables: prim_adv_int_min: Adv_Interval, prim_adv_int_max: Adv_Interval, prim_adv_phy: PHY, sec_adv_max_skip: Secondary_Advertising_Max_Skip, sec_adv_phy: PHY
4.1.5.2 SCN

The common initial states of the scanning state.

State: Passive Scanning

The HCI commands to set the IUT to scan in the passive mode with the parameters filtering policy, scan interval, and scan window are displayed in the sequence chart below.

![Figure 4.12: Passive Scanning Preamble Steps](image1)

State: Active Scanning

The sequence chart below displays the HCI commands to set the IUT to scan in the active mode with the parameters of filtering policy, scan interval and scan window.

![Figure 4.13: Active Scanning Preamble Steps](image2)
**State: Low Duty Cycle Directed Advertising**

The HCI commands to set the IUT to advertising using Low Duty Cycle Directed Advertising with parameters are displayed in **Figure 4.14**.

**misc variables** 
- $\text{adv\_interval\_min}$: $\text{Adv\_Interval}$, $\text{adv\_interval\_max}$: $\text{Adv\_Interval}$, $\text{Adv\_type}$: 0x04,
- $\text{own\_address\_type}$: $\text{Address\_Type}$, $\text{direct\_address\_type}$: $\text{Address\_Type}$, $\text{adv\_channel\_map}$: $\text{Adv\_Channel\_Map}$

**Figure 4.14: Low Duty Cycle Direct Advertising Preamble Steps**
4.1.5.3 INI

The common initial states of the initiating state.

State: Initiating

The HCI commands to set the IUT to initiate a connection with parameters are displayed in Figure 4.15.

Figure 4.15: Initiating Preamble Steps

Note: peer_address is an address used by the Lower Tester.
State: Connection Setup White Listed

```
Connection Setup White Listed Preamble Steps

Connection Setup White Listed

HCI_LE_Clear_Device_White_List

HCI_Command_Complete
(0x00)

HCI_LE_Add_Device_White_List
(device_addresses[], address_types[])

HCI_Command_Complete
(0x00)

Connection Setup White Listed

Step 2

Repeat Step 2 for all ‘i’ in ‘device_addresses[] & address_types[]’
```

Figure 4.16: Connection Setup White Listed Preamble Steps

The initiator device filtering uses only the white listing mechanism.

4.1.5.4 SLA

The common initial states of the slave role.

State: Connected Slave

The HCI commands to set the IUT to accept a connection and start maintaining it in the slave role with parameters are displayed in Figure 4.17.

```
Connected Slave Preamble Steps

Buffer Size Read

Undirected Advertising Preamble Steps (no scan response data set)

Undirected Advertising

Accepting Connections Test Procedure Steps

Connected Slave
```

Figure 4.17: Connected Slave Preamble Steps
The preamble steps for undirected advertising require the parameters to first start undirected advertising, then for the Lower Tester to initiate the connection setup to the IUT. The parameter values supplied to the IUT for the advertising are not significant in these tests, but the connection parameters may be. The test procedure steps for accepting connections are performed once in these preamble steps using the parameter values, in order to complete the connection setup procedure.

State: Slave Connection Terminated

In each test where a slave device is connected but connection termination is not a part of the test objective, a postamble is added to the test procedure steps. The objective with the postamble steps is to bring the IUT to the standby state.

![Diagram of Slave Connection Terminated Postamble Steps]

The asterisk denoting the packet type in step 1 above means that only the acknowledgement scheme is followed in the steps and the packet types are not observed.
State: Slave Connected on Secondary Advertising Physical Channel

The HCI commands to connect the IUT to the Lower Tester on a secondary advertising channel with parameters as displayed in the sequence chart below.

![Sequence Chart]

**Figure 4.19: Slave Connected on Secondary Advertising Physical Channel Preamble Steps**

Extended Advertising msc variables: prim_adv_int_min: Adv_Interval, prim_adv_int_max: Adv_Interval, prim_adv_phy: PHY, sec_adv_max_skip: Secondary_Advertising_Max_Skip, sec_adv_phy: PHY
4.1.5.5 MAS

The common initial states of the master role.

**State: Connected Master**

The HCI commands to have the IUT maintain a connection in the master role with parameters are displayed in the sequence chart below.

![Sequence chart](image)

*Figure 4.20: Connected Master Preamble Steps*

The preamble steps for initiating require the parameters, but every value may not be significant in the tests for the connected master. The test procedure steps for connection initiation are performed once in these preamble steps, in order to complete the connection setup.
State: Master Connection Terminated

In each test where a master device is connected but connection termination is not a part of the test objective, a postamble is added to the test procedure steps. The objective with the postamble steps is to bring the IUT to the standby state.

The asterisk denoting the packet type in step 1 above means that only the acknowledgement scheme is followed in the steps and the packet types are not observed.
State: Master Connected on Secondary Advertising Physical Channel

The HCI commands to connect the IUT to the Lower Tester on a secondary advertising channel with parameters as displayed in the sequence chart below.

Figure 4.22: Master Connected on Secondary Advertising Physical Channel Preamble Steps

Master Connected on Secondary Advertising Physical Channel msc variables: init_phys: PHY, scan_int[]: Scan_Interval, scan_win[]: Scan_Window, conn_int_mn: Connection_Interval, conn_int_mx: Connection_Interval, conn_latency[]: Conn_Latency, sup_timeout[]: Supervision_Timeout, mn_ce_len[]: Minimum_CE_Length, mx_ce_len[]: Maximum_CE_Length
4.1.5.6 SEC

State: Encrypted Address Calculated

The random address generation behavior is an extract from GAP [1] Section 2.1.2 and represents here the typical HCI sequences required from a Controller. The identity resolving key ‘irk’ is used in the test procedures in group ‘SEC’.
State: Encryption Keys Calculated

Encryption Keys Calculated Preamble Steps

```
State: Encrypted Slave Connection (ir: Key, er: Key, adv_interval_min: Adv_Interval, adv_interval_max: 
Adv_Interval, address_type: Address_Type, adv_channel_map: Adv_Channel_Map, conn_interval: 
Conn_Interval, conn_latency: Conn_Latency, conn_timeout: Conn_Timeout, sca: SCA)

State: Encrypted Master Connection (ir: Key, er: Key, scan_interval: Scan_Interval, scan_window: 
Scan_Window, address_type_peer: Address_Type, peer_address: Peer_Address, address_type_own: 
Address_Type, conn_interval: Conn_Interval, conn_latency: Conn_Latency, conn_timeout: 
Conn_Timeout

Encryption keys are input to a Controller from [1] (part H, section 2.4.2). The keys to generate random 
addresses are obtained from the identity root "IR". The IR is referred to as 'ir' and has the default value 
0x1123456789AABBCCDDEFF00. The encryption root, ER, is referred to as 'er' and has the same 
default value. The key to encrypt LL data is referred to as 'ltk'.
```
4.1.5.7 Common Test Procedure Steps

The common Test Procedure steps section describes alternative optional test steps that a device may invoke at will during the execution of Test Procedure steps when the test would otherwise end with an inconclusive verdict.

4.1.5.7.1 Recovery Actions in Test Steps

The test procedure steps typically outline the correct behavior that leads to execute the test procedure successfully, but is not a full implementation description for a test system. In the evaluation of the IUT behavior against a test procedure description, some provision is needed for failures in a testing configuration with a test system and an IUT. Below are sample behaviors that may occur in execution of a test procedure:

- **Step for expecting a packet, but receiving none:**
  Repeat the step expecting the packet a number of times to reflect the confidence in the verdict to assign. The test procedure may fail to execute after the repetition count or time interval, or a Fail verdict may be assigned with a total number of missing packets.

- **Step for expecting a packet, but receiving a different packet with a correct CRC:**
  If the packet is from the IUT and matches the contents expected (advertising channels) or the timing and synchronization word (data channels), assume the IUT has sent the packet and assign a Pass verdict.
  If there are no alternatives for the test procedure step and the testing limitations do not describe otherwise, assign a Fail verdict.

- **Step for expecting a packet and receiving one with an incorrect CRC:**
  If the packet is from the IUT and matches the timing, repeat the step expecting the packet a number of times to reflect the confidence in the verdict to assign (Assuming that a packet from the IUT is corrupted). Stop the test procedure execution after the repetition count.

- **Step for expecting no packet, but receiving one with a correct CRC:**
  If the packet received is from the IUT fail the test procedure step (Assuming the IUT has sent an inopportune packet).

- **Step for expecting no packet, but receiving one with an incorrect CRC:**
  If the packet is from the IUT and matches the timing, fail the test procedure step (Assuming the IUT has sent an inopportune packet).

4.1.5.7.2 Test Step Repetition Count

In steps repeating a test procedure in order to assign a verdict, the phrase ‘a number of times’ is used. The actual repetition count is based on the failure or success probability of the particular protocol function that is executed in the test. In the case of a particular operation (e.g. correct one-way reception, transmission and reception, and cetera) the repetition count is based on the bit error rate verified for an IUT in the RF PHY tests [7].

The confidence level is proportional to the repetition count of the test.
4.1.5.7.3 Optional Test Steps

Note that the IUT may use these procedures in a way that the resulting trace of the Test Procedure does not exactly match the test description alone; however, taking into account the optional tests steps, it may still match the verdict criteria.

- Connection Update and Connection Parameters Request Procedures Optional Test Steps

![Diagram](https://via.placeholder.com/150)

Figure 4.25: Connection Update and Connection Parameters Request Procedure Optional Test Steps

The IUT may request a change to the connection parameters, using either the Connection Update Procedure or the Connection Parameters Request Procedure, only if:

- the test does not mention the Connection Update Procedure or the Connection Parameters Request Procedure, or
- the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.
• **Channel Map Update Procedure Optional Test Steps**

![Diagram of Channel Map Update Procedure]

*Figure 4.26: Channel Map Update Procedure Optional Test Steps*

The IUT may request a channel map update when master only if:
- the test does not mention the Channel Map Update Procedure, or
- the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.

• **Version Exchange Procedure Optional Test Steps**

![Diagram of Version Exchange Procedure]

*Figure 4.27: Version Exchange Procedure Optional Test Steps*

The IUT or Lower Tester may initiate a Version Exchange Procedure before or during the Test Procedure as long as the test does not mention this procedure in the Test Procedure.

The Lower Tester may initiate this procedure to reset/flush any version information that the IUT may have cached from a previous connection with it.
• Feature Exchange Procedure Optional Test Steps

The IUT or Lower Tester may initiate a Feature Exchange Procedure before or during the Test Procedure as long as the test does not mention this procedure in the Test Procedure.

The Lower Tester may initiate this procedure to reset/flush any feature information that the IUT may have cached from a previous connection with it.
- **Data Length Update Procedure Optional Test Steps**

  ![Diagram of Data Length Update Procedure](image)

  Figure 4.29: Data Length Update Procedure Optional Test Steps

  The IUT may initiate a Data Length Update Procedure only if:
  - the test does not mention the Data Length Update Procedure, or
  - the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.

- **LE Ping Procedure Optional Test Steps**

  ![Diagram of LE Ping Procedure](image)

  Figure 4.30: LE Ping Procedure Optional Test Steps

  IUT may initiate an LE Ping Procedure only if:
  - the test does not mention the LE Ping Procedure, or
  - the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.
• LE PHY Update Procedure Optional Test Steps

Figure 4.31: LE PHY Update Procedure Optional Test Steps

If the test procedure involves the Upper Tester issuing an HCI_LE_Set_PHY command that is not intended to change the PHY (e.g., one with ALL_PHYS = 0x03), then the IUT may optionally initiate a PHY Update Procedure which is not described in the test procedure by issuing an LL_PHY_REQ PDU to the Lower Tester. If the Lower Tester is the master, it shall respond with an LL_PHY_UPDATE_IND that does not change either PHY. If the Lower Tester is the slave, it shall respond with an LL_PHY_RSP PDU that specifies the current PHYs in both directions. In the latter case, if the IUT does not respond with an LL_PHY_UPDATE_IND that does not change either PHY, assign a Fail verdict. The HCI_LE_PHY_Update_Complete event in the test procedure always follows this optional procedure.
- **LE PHY Update Procedure (LE Coded Switch) Optional Test Steps**

![Diagram](https://via.placeholder.com/150)

**Figure 4.32: LE PHY Update Procedure (LE Coded Switch) Optional Test Steps**

If the test procedure involves the Upper Tester issuing an HCI_LE_Set_PHY command that is intended to change the PHY to or from LE Coded PHY, a change to either the maximum Payload length or the maximum transmission time of packets for the IUT may occur. Optionally, the IUT may initiate a Data Length Update Procedure after the PHY change. If both or either of the cases occur, the IUT notifies the Upper Tester with at least one HCI_LE_Data_Length_Change event before or after the HCI_LE_PHY_Update_Complete event.
• Test Command Generated Isochronous SDUs Optional Test Steps

An isochronous IUT may utilize an Upper Tester that meets one of the following conditions:

1. The Upper Tester cannot provide isochronous data SDUs to the IUT.
2. The Upper Tester and/or its associated HCI transport do not support sufficient bandwidth to support transmission of isochronous SDUs in a given isochronous test procedure.

In such cases, it is permissible for the Upper Tester to utilize the HCI_LE_ISO_Transmit_Test command to generate the SDUs required for a given test procedure as shown in Figure 4.33. The transmit test command is terminated at the end of the test procedure as shown.

The SDUs generated by the ISO Transmit Test command have a specific, fixed format that cannot be altered by the Upper Tester. Some test procedures call out specific payload contents to confirm correct operation. In those cases, a Lower Tester may need to analyze the contents of any received PDU{s} generated using the ISO Transmit Test command differently than defined in a given isochronous test procedure.

Figure 4.33: Test Command Generated Isochronous SDUs Optional Test Steps
Test Command Received Isochronous SDUs Optional Test Steps

An isochronous IUT may utilize an Upper Tester that meets one of the following conditions:

1. The Upper Tester cannot receive isochronous data SDUs from the IUT.
2. The Upper Tester and/or its associated HCI transport do not support sufficient bandwidth to support reception of isochronous SDUs in a given isochronous test procedure.

In such cases, it is permissible for the Upper Tester to utilize the HCI_LE_ISO_Receive_Test and HCI_LE_ISO_Read_Test_Counters commands to receive the SDUs required for a given test procedure as shown in Figure 4.34. The receive test command is terminated at the end of the test procedure as shown.

As the actual contents of the isochronous data cannot be processed by the Upper Tester, the received data cannot be considered for pass/fail criteria, only the successful or unsuccessful reception of the data as defined in a given isochronous test procedure.

Figure 4.34: Test Command Received Isochronous SDUs Optional Test Steps
• **BIG Termination Procedure Optional Test Steps**

![Diagram](image)

Figure 4.35: BIG Termination Optional Test Steps

If the test procedure mandates that an IUT in the isochronous broadcaster role terminate its BIG, the IUT must provide six consecutive BIG_TERMINATE_IND PDUs prior to the instant when the BIG is to be terminated. Additional BIG_TERMINATE_IND PDUs may be broadcast prior to the instant, though they need not be consecutive.

• **Minimum Number of Used Channels Procedure Optional Test Steps**

![Diagram](image)

Figure 4.36: Minimum Number of Used Channels Procedure Optional Test Steps

The IUT may request a change in the minimum number of channels be used on the indicated PHY, only if:

- The test does not mention a Minimum Number of Used Channels Procedure, or
- The procedure is carried out before the first step of the Test Procedure, and the Initial Conditions mentioned in the test are still respected.
4.1.5.8 Pass/Inconclusive/Fail Verdict Conventions

Each test case has an Expected Outcome section, which outlines all the detailed pass criteria conditions that shall be met by the IUT to merit a Pass Verdict.

Certain test cases also have an Inconclusive Verdict defined. If the conditions for this verdict are met, the test provides evidence that the IUT neither meets nor violates the test case; instead it means that the test case was not applicable to the IUT, and therefore a Pass Verdict is not required in order to achieve Qualification of the IUT. Implementers are encouraged to provide mechanisms to avoid the behavior leading to an Inconclusive condition during testing.

The convention in this test suite is that, unless there is a specific set of fail conditions outlined in the test case, then the IUT fails the test case as soon one of the pass criteria conditions or the inconclusive conditions (where they exist) cannot be met, and if this occurs, the outcome of the test shall be the Fail Verdict.

For an Inconclusive Verdict, all the pass criteria conditions apply up to the point in the test procedure where an Inconclusive Verdict is identified. If one of the pass criteria in a step prior to the Inconclusive Verdict cannot be met, the outcome of the test shall be the Fail Verdict and not the Inconclusive Verdict.

4.1.5.9 ISO

State: Connected Isochronous Stream, Master, Test

The sequence for the IUT as Master to establish a CIS with the Lower Tester as Slave using the HCI_LE_Set_CIG_Parameters_Test command.
Figure 4.37: Connected Isochronous Stream, Master, Connection Test Steps

Connected Isochronous Stream, Master, Test msc variables: sdu_int_m2s: SDU_Interval, sdu_int_s2m: SDU_Interval, ft_m2s: Flush_Timeout, ft_s2m: Flush_Timeout, iso_int: ISO_Interval, packing: Packing, framing: Framing, cis_cnt: CIS_Count, nse[]: NSE, mx_sdu_m2s[]: Data_Len, mx_sdu_s2m[]: Data_Len, mx_pdu_m2s[]: Data_Len, mx_pdu_s2m[]: Data_Len, phy_m2s[]: PHY, phy_s2m[]: PHY, bn_m2s[]: Burst_Number, bn_s2m[]: Burst_Number
**State: Connected Isochronous Stream, Slave**

The sequence for the IUT as Slave to accept a CIS with the Lower Tester as Master.

**Lower Tester**

Connected Isochronous Stream, Slave

**IUT**

**Upper Tester**

For Each CIS Request

<table>
<thead>
<tr>
<th>LL_CIS_REQ</th>
<th>LL_CIS_IND</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HCI_LE_CIS_Request Event</strong></td>
<td><strong>HCI_LE_CIS_Indication</strong></td>
</tr>
<tr>
<td>(acl_cnct_hdnl, cis_cnct_hdnl, cig_id, cis_id)</td>
<td>(Status: 0x00)</td>
</tr>
</tbody>
</table>

**HCI_LE_CIS_Established event**

| Status: 0x00, cnct_hdnl, cig_sync_dly, cis_sync_dly, trans_lncy, phy_m2s, phy_s2m |

**HCI_LE_Setup_ISO_DataPath**

| cnct_hdnl, 0x01, 0x01 |

**HCI_Command_Complete**

| Status: 0x00, cnct_hdnl |

**ACL Connection Established. IUT is Slave.**

**Figure 4.38: Connected Isochronous Stream, Slave Accept Connection Test Steps**

Connected Isochronous Steam, Slave, Lower Tester variables: sdu_int_m2s: SDU_Interval, sdu_int_s2m: SDU_Interval, ft_m2s: Flush_Timeout, ft_s2m: Flush_Timeout, iso_int: ISO_Interval, packing: Packing, framing: Framing, cis_cnt: CIS_Count, nse[]: NSE, mx_sdu_m2s[]: Data_Len, mx_sdu_s2m[]: Data_Len, mx_pdu_m2s[]: Data_Len, mx_pdu_s2m[]: Data_Len, phy_m2s[]: PHY, phy_s2m[]: PHY, bn_m2s[]: Burst_Number, bn_s2m[]: Burst_Number
State: Isochronous Broadcasting, Test

The sequence for the IUT to enter the Isochronous Broadcasting State using the HCI_LE_Create_BIG_Test command.

**Figure 4.39: Isochronous Broadcasting, Test Steps**

State: Synchronized to a Broadcast Isochronous Stream

The sequence for the IUT in the Synchronization state to synchronize to a Broadcast Isochronous Stream.

![Diagram of synchronization process]

**Figure 4.40: Synchronized to a Broadcast Isochronous Stream, Test Steps**


### 4.1.6 Data Fragmentation over HCI

When a test case involves data being sent from the Lower Tester to the IUT and then reported to the Upper Tester, or vice versa, then the following requirements shall be met.

- The data bytes received shall have the same values, in the same order, as those sent.
- The number and location of "start flags" within the received data shall be the same as those in the sent data. A “start flag” occurs immediately before the first byte of a Link Layer packet with LLID set.
to 0x02 or an HCI data packet with Packet_Boundary_Flag set to 0x00 or 0x02, and nowhere else. Note that packets containing no data could result in more than one start flag at the same location.

Provided that these requirements are met, the specific fragmentation of the data is not part of the pass criteria for a test case.

4.1.7 Data PDUs and Empty PDUs
The test procedures and message sequence charts of tests may expect empty packets or data PDUs from and to the IUT. In these places, any valid data channel PDU shall also be permitted, provided that the contents of data packets conform to section 4.1.6 and that control packets only appear where permitted by section 4.1.5.7.3 and not forbidden by the Core Specification (e.g. during a conflicting ongoing procedure).

4.1.8 Outstanding Commands Prior to Disconnection
In test cases where the IUT is in either the master or slave role, if there are outstanding commands relating to the connection, and the connection gets disconnected (the disconnection may be initiated either by the IUT or the Lower Tester), then the Upper Tester ensures that:

- If the IUT completes those outstanding commands, then it does that with a non-zero status and before returning the HCI_Disconnection_Complete event.
- The IUT does not send any events for that connection handle after sending the HCI_Disconnection_Complete event.

4.2 DDI
Test of the device discovery procedures.

4.2.1 Common PDU Contents
The packet descriptions for advertising channel packets sent and accepted by the Lower Tester are displayed below. The addresses used in tests vary for the Lower Tester, in case of the IUT the address is expected to match the registered IXIT value. The data used in tests varies from no advertising data to data with different length and content.

4.2.1.1 Legacy Advertising
ADV_NONCONN_IND PDU:

<table>
<thead>
<tr>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb</td>
<td>msb</td>
</tr>
<tr>
<td>Type</td>
<td>RFU</td>
</tr>
<tr>
<td>'0100'</td>
<td>'00'</td>
</tr>
</tbody>
</table>
### ADV_IND PDU:

<table>
<thead>
<tr>
<th>Field</th>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb</td>
<td>msb</td>
<td>lsb</td>
</tr>
<tr>
<td>Type</td>
<td>'0000'</td>
<td>LSO MSO</td>
</tr>
<tr>
<td>RFU</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>ChSel</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>TxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>RFU</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>(8 bits)</td>
<td>AdvA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6 octets)</td>
</tr>
</tbody>
</table>

### ADV_DIRECT_IND PDU:

<table>
<thead>
<tr>
<th>Field</th>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb</td>
<td>msb</td>
<td>lsb</td>
</tr>
<tr>
<td>Type</td>
<td>'1000'</td>
<td>LSO MSO</td>
</tr>
<tr>
<td>RFU</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>ChSel</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>TxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>RxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>'00110000'</td>
<td>AdvA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6 octets)</td>
</tr>
</tbody>
</table>

### ADV_SCAN_IND PDU:

<table>
<thead>
<tr>
<th>Field</th>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb</td>
<td>msb</td>
<td>lsb</td>
</tr>
<tr>
<td>Type</td>
<td>'0110'</td>
<td>LSO MSO</td>
</tr>
<tr>
<td>RFU</td>
<td>'00'</td>
<td></td>
</tr>
<tr>
<td>TxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>RFU</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>(6 bits)</td>
<td>RFU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'00'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AdvA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6 octets)</td>
</tr>
</tbody>
</table>

### SCAN_REQ PDU:

<table>
<thead>
<tr>
<th>Field</th>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb</td>
<td>msb</td>
<td>lsb</td>
</tr>
<tr>
<td>Type</td>
<td>'1100'</td>
<td>LSO MSO</td>
</tr>
<tr>
<td>RFU</td>
<td>'00'</td>
<td></td>
</tr>
<tr>
<td>TxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>RxAdd</td>
<td>'0'</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>'001100'</td>
<td>RFU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'00'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ScanA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6 octets)</td>
</tr>
</tbody>
</table>
SCAN_RSP PDU:

<table>
<thead>
<tr>
<th>Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isb msb</td>
<td>Isb msb</td>
</tr>
<tr>
<td></td>
<td>Isb msb</td>
</tr>
<tr>
<td>Type '0010'</td>
<td>RFU '00'</td>
</tr>
</tbody>
</table>

4.2.1.2 Extended Advertising

In all tests where the IUT is transmitting ADV_EXT_IND and possibly AUX_ADV_IND extended advertising PDUs to the Lower Tester, the Lower Tester shall make the following checks which shall form part of the Pass Verdict:

- If the Upper Tester specified anonymous advertising then neither the ADV_EXT_IND PDU nor the AUX_ADV_IND PDU shall include an AdvA field. Otherwise exactly one shall include an AdvA field. (Note: the Upper Tester will not specify anonymous advertising if no, or zero length, advertising data is specified.)

- If the Upper Tester specified directed advertising then either the ADV_EXT_IND PDU or the AUX_ADV_IND PDU, but not both, shall include a TargetA field. Otherwise neither shall include a TargetA field.

- If an ADV_EXT_IND PDU has an AdvMode field containing 01b (connectable) or 10b (scannable), it shall not include an AdvA or TargetA field.

- If an ADV_EXT_IND PDU is sent on the LE Coded PHY and includes an AuxPtr field, it shall not include an AdvA or TargetA field.

- If a PDU includes an AdvA field, it shall contain the IUT's Advertising Address specified by the Upper Tester during the test procedure.

- If a PDU includes a TargetA field, it shall contain the Lower Tester's Device Address specified by the Upper Tester during the test procedure.

4.2.2 ADV

Tests that the IUT behaves according to the device discovery procedures in the advertiser role.

4.2.2.1 Common PDU Contents

The advertising channel packet contents that are sent by the Upper Tester sent or expected by the Lower Tester to be received are defined in '/LL/DDI'.

4.2.2.2 LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events]

- Test Purpose
  
  Test that an advertiser IUT sends the advertising packets of non-connectable event type, with correct contents and with correct event timing.
The Lower Tester observes the event timing and packet contents on the selected advertising channel.

- **Reference**
  
  [4] 4.4.2.6, 4.4.2.2

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State: Non-Connectable Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)

- **Test Procedure**

  ![Test Procedure Diagram](image)

  - **ADV_NONCONN_IND**
    
    - **ADV_NONCONN_IND**
      
      - **ADV_NONCONN_IND**
        
        - **ADV_NONCONN_IND**
          
          - **ADV_NONCONN_IND**
            
            - **ADV_NONCONN_IND**
              
              - **ADV_NONCONN_IND**

  **Figure 4.41:** LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events]
3. Configure Lower Tester to monitor advertising packets from the IUT.
4. Upper Tester enables non-connectable advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
5. Expect the IUT to send ADV_NONCONN_IND on the selected advertising channel.
6. Expect the following event to start one advertising interval after the start of the first packet.
7. Repeat steps 3–4 until a number of advertising intervals (100) have been detected.
8. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

- **Expected Outcome**

  **Pass Verdict**

  The test procedure completes using the selected advertising interval for all supported advertising channels.

  The timing range detected for advertising events is from TSPX_adv_interval_min to TSPX_adv_interval_min + 10.0 ms (calculated for the minimum advertising interval specified in [9]).

- **Notes**

  The required accuracy for advertising event intervals is a microsecond, from the interval setting granularity of 625 µs. The delay applied is regarded as a range of up to 10.000 ms, making the timing observation a question of whether the measured range is within the advertising interval plus any delay selected. Taking into account the clock drift of 500 ppm of the low power mode at the minimum advertising interval, tens of microseconds are affected in the accuracy. Jitter for low power mode (16 µs) has the same effect on the accuracy. The clock drift would affect tens of milliseconds at the maximum advertising interval. The 16µs requirement applies to the advertising and periodic advertising intervals, the advDelay value, all intervals between packets in the same extended advertising event or periodic advertising event, and all offsets specified by the AuxPtr and SyncInfo fields of advertising PDUs.

  The minimum actual time interval between any two events required is therefore the interval specified plus negative drift plus negative jitter: 

  \[(0.625 \text{ ms} \times \text{Adv\_Interval}) - (0.625 \text{ ms} \times \text{Adv\_Interval} \times 500 \text{ ppm}) - 0.032 \text{ ms}\] from the previous event.

  The maximum actual time interval is

  \[(0.625 \text{ ms} \times \text{Adv\_Interval} + 10.000 \text{ ms}) \times (1 + 500 \text{ ppm}) + 0.032 \text{ ms}\] from the previous event.

  Since repeated measurements close to the maximum advertising interval of seconds is not feasible because of test time accumulation (over a minute to repeat 10 times), the accuracy of timing is tested with supported values in the lower interval range.

  The order of deviation allowed as result of drift and jitter is the minimum measurement accuracy. Measurement results are rounded to the next decimal after calculations for comparison with the requirements. For the maximum advertising interval, this gives accuracy of 0.01 s and for the minimum advertising interval 0.1 ms.

  The total time the test procedure is attempted is based on the probability of detecting 2 consequent advertising events and resolves to around 100 events, after which the test procedure is stopped.

4.2.2.3 **LL/DDI/ADV/BV-02-C [Undirected Advertising Events]**

- **Test Purpose**

  Test that an advertiser IUT sends advertising packets of an event with correct contents on a selected advertising channel with correct event timing.
The Lower Tester observes the packet and event timing and packet contents on the selected advertising channel.

- Reference
  [3] 4.4.2.3

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, length of device name used, common device name)

- Test Procedure

  ![Diagram of test procedure](Image)

  **Figure 4.42: LL/DDI/ADV/BV-02-C [Undirected Advertising Events]**
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester enables undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
3. Lower Tester expects the IUT to send ADV_IND packets on the selected advertising channel.
4. Expect the next event to start after advertising interval time calculated from the start of the first packet.
5. Repeat steps 3–4 until a number advertising intervals (100) have been detected.
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

- Expected Outcome

  **Pass Verdict**
  The test procedure completes using the selected advertising interval.
  The test procedure completes the selected advertising channel.
  The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms.

- Notes
  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.4 **LL/DDI/ADV/BV-03-C [Advertising Data: Non-Connectable]**

- Test Purpose
  Test that an advertiser IUT sends advertising packets of a non-connectable event type with data on a selected advertising channel.
  The Upper Tester submits data to the IUT and the Lower Tester observes the IUT including data to the selected advertising packets on the advertising channel.

- Reference
  [3] 4.4.2.6

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- Test Procedure
  Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using the selected advertising channel, with default data lengths of 1 and 31. If the IUT has a maximum data length of less than 31 bytes, this may be specified via an IXIT value.
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester configures non-connectable advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
3. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and receives an HCI_Command_Complete in response. The data element used in the command is the length of the data field. The data length is 1 byte.
4. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.
5. Lower Tester expects the IUT to send ADV_NONCONN_IND packets including the data submitted in step 3 starting an event on the selected advertising channel.

6. Expect the following event to start after advertising interval time calculating from the start of the first packet.

7. Repeat steps 5–6 until a number of advertising intervals (50) have been detected.

8. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising function and receives an HCI_Command_Complete event in response.

9. Upper Tester sends an HCI_LE_Set_Advertising_Data to configure the IUT to send advertising packets without advertising data and receives an HCI_Command_Complete event in response.

10. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.

11. Lower Tester expects the IUT to send ADV_NONCONN_IND packets including no advertising data starting an event on the selected advertising channel.

12. Expect the next event to start after advertising interval time calculating from the start of the first packet.

13. Repeat steps 11–12 until a number of advertising intervals (50) have been detected.

14. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.

15. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and receives an HCI_Command_Complete in response. The data element is a number indicating the length of the data field in the first octet encoded unsigned least significant bit first and the rest of the octets zeroes. The data length is either 31 bytes by default or it may be specified by IXIT value if less than 31 bytes.


**Expected Outcome**

**Pass Verdict**

The test procedure executes with the IUT advertising using non-connectable event type.

The IUT transmits data as submitted in the HCI commands.

**4.2.2.5 LL/DDI/ADV/BV-04-C [Advertising Data: Undirected]**

**Test Purpose**

Tests that an advertiser IUT sends advertising packets of an undirected type of event with data on a selected advertising channel.

The Upper Tester submits data to the IUT and the Lower Tester observes the IUT including data to the advertising packets on the selected advertising channel.

**Reference**

[3] 4.4.2.3

**Initial Condition**

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, selected type of advertising events, supported type of own address, selected advertising channel map)
Test Procedure

Execute the test procedure using undirected advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using the selected advertising channel, with data lengths of 1 and 31.

1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester configures undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
3. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and receives an HCI_Command_Complete in response. The data element used in the command is the length of the data field. The data length is 1 byte.

4. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.

5. Lower Tester expects the IUT to send ADV_IND packets including the data submitted in step 3 starting an event on the selected advertising channel.

6. Expect the following event to start after advertising interval time calculating from the start of the first packet.

7. Repeat steps 5–6 until a number of advertising intervals (50) have been detected.

8. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising function and receives an HCI_Command_Complete event in response.

9. Upper Tester sends an HCI_LE_Set_Advertising_Data to configure the IUT to send advertising packets without advertising data and receives an HCI_Command_Complete event in response.

10. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.

11. Lower Tester expects the IUT to send ADV_IND packets including no advertising data starting an event on the selected advertising channel.

12. Expect the next event to start after advertising interval time calculating from the start of the first packet.

13. Repeat steps 11–12 until a number of advertising intervals (50) have been detected.

14. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.

15. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and receives an HCI_Command_Complete in response. The data element is a number indicating the length of the data field in the first octet encoded unsigned least significant bit first and the rest of the octets zeroes. The data length is 31 bytes.


- Expected Outcome

Pass Verdict

The test procedure executes with the IUT advertising using undirected connectable event type, The IUT transmits data as submitted in the HCI commands.

4.2.2.6 LL/DDI/ADV/BV-05-C [Scan Request: Undirected Connectable]

- Test Purpose

Tests that an advertiser IUT responds to a scan request and continues advertising after the response.

The Lower Tester requests information from the IUT, receives a response, then checks that the advertising resumes.

- Reference

[3] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State for all IUTs: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, length of device name used, selected
name) AND White List All Unknown Devices (Allow Scan Request from Any, Allow Connect Request from Any (0x00)).

- Test Procedure

Execute the test procedure using the selected advertising interval.

1. Upper Tester configures undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a common public device address as parameter.
3. Configure Scan Response Data in the IUT using device name length of 0 as response data.
4. Lower Tester sends a SCAN_REQ packet on the selected advertising channel after receiving an ADV_IND packet from IUT on the advertising channel configured in step 3. The SCAN_REQ is sent T_IFS after the end of an ADV_IND packet.
5. Lower Tester receives a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
6. Repeat steps 4–5 30 times or until IUT sends a SCN_RSP.
7. Configure Scan Response Data in the IUT using device name length of 31 as response data.
8. Repeat steps 4–6.
9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a public device address that differs from the IUT address in the most significant octet.
10. Configure Scan Response Data in the IUT using device name length of 0 as response data.
11. Repeat steps 4–6.
12. Configure Scan Response Data in the IUT using device name length of 31 as response data.
13. Repeat steps 4–6.
14. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a public device address that differs from the IUT address in the most and least significant octets.
15. Repeat steps 4–6.
16. Configure Scan Response Data in the IUT using device name length of 31 as response data.
17. Repeat steps 4–6.

• Expected Outcome
  Pass Verdict
  The test procedure completes using the selected advertising interval.
  The test procedure completes with the IUT responding on each advertising channel.
  The IUT responds in each case of different scanner address used.
  The timing deviations detected for packets in active mode are within the 2 µs range around T_IFS.

• Notes
  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.7 LL/DDI/ADV/BV-06-C [Connection Request]

• Test Purpose
  Tests that an advertiser IUT receives a connection request and stops advertising after its reception.
  The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.

• Reference
  [3] 4.4.2.3, 4.3.2

• Initial Condition
  Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map.
  State for all IUTs: Undirected Advertising (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name) AND White List All Unknown Devices (Allow Scan Request from Any, Allow Connect Request from Any (0x00)).
1. Upper Tester enables undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
2. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the first supported advertising channel.
3. Configure Lower Tester to use a white public device address as parameter of CONNECT_IND.
4. The Lower Tester receives an ADV_IND packet from the IUT and responds with a CONNECT_IND packet after the end of the advertising packet.
5. The Lower Tester receives no ADV_IND packet after the advertising interval from the IUT. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 4 and as postamble: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
7. Configure Lower Tester to use a public device address that differs from the IUT address in the most significant octet as parameter of CONNECT_IND.
8. Repeat steps 4–6.
9. Configure Lower Tester to use a public device address that differs from the IUT address in the least significant octet as parameter of CONNECT_IND.
10. Repeat steps 4–6.
11. Configure Lower Tester to use a public device address that differs from the IUT address in the most and least significant octets as parameter of CONNECT_IND.
12. Repeat steps 4–6.
13. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the second supported advertising channel.
14. Repeat steps 3–12.
15. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the third supported advertising channel.
16. Repeat steps 3–12.

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels,

The IUT reports the requested connection with an HCI event for all address variants applied.

• Notes

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.8 LL/DDI/ADV/BV-07-C [Scan Request Connection Request]

• Test Purpose

Test that the IUT accepts a scan request immediately followed by a connection request.

The Lower Tester first acts in the active scanning state sending scan requests to the IUT, then after receiving a scan response from the IUT, it switches to initiating state to send a connection request to the IUT.

• Reference

[3] 4.4.2.3

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name).
• **Test Procedure**

1. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, a selected advertising interval between the minimum and maximum advertising intervals, and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)’.

2. Upper Tester sends an **HCI_LE_Set_Advertising_Parameters** command with parameters (Conn. Undir, Public Addr., Policy: 0x00) and receives an **HCI_Command_Complete** event from the IUT.

---

**Figure 4.47: LL/DDI/ADV/BV-07-C [Scan Request Connection Request]**

- **Scan Request** (scanner_address)
- **Scan Response** (data:"IUT")
- **Connect Indication** (initiator_address)
- **Connect Complete** (Status: 0x00)
3. Configure Lower Tester to monitor the advertising, scan response and connection procedures of the IUT, sending a SCAN_REQ and a CONNECT_IND packet on a supported advertising channel (defined as an IXIT).
4. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with an SCAN_REQ packet on the selected advertising channel T_IFS after the end of an advertising packet.
5. Lower Tester receives an SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
6. Repeat steps 4–5 30 times or until IUT sends SCAN_RSP.
7. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
8. The Lower Tester receives no ADV_IND packet after advertising interval from the IUT after sending the connection request to indicate that the IUT has stopped advertising.
9. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT.
10. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 9).

- Expected Outcome

Pass Verdict

The test procedure completes when the IUT responds to the scan request and connection is successful.

4.2.2.9 LL/DDI/ADV/BV-08-C [Scan Request Device Filtering]

- Test Purpose

Tests that an advertiser IUT filters scanners according to the white list and filtering policy set.

The Lower Tester transmits scan requests to the IUT using addresses and address types that either pass or fail the filter, then observes the response from the IUT on the advertising channels used.

- Reference

[3] 4.3.2, 4.4.2.3

- Initial Condition

Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map.

State for the first execution of the test procedure: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, black list all unknown devices).

State for the second execution of the test procedure: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, allow connection requests from unknown devices).
• Test Procedure

1. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.

2. Upper Tester sends an HCI_LE_Set_Scan_Response_Data command with data set to “IUT” and receives an HCI_Command_Complete event from the IUT.

3. Lower Tester address type is set to Public Address type.

Figure 4.48: LL/DDL/ADV/BV-08-C [Scan Request Device filtering]
4. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send an SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address that differs from the IUT address in the least significant octet (an address black listed in the policy applied).

5. Lower Tester receives an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with the selected address on the selected advertising channel T_IFS after the end of an advertising packet.

6. Lower Tester receives no response from the IUT.

7. Repeat steps 5–6 30 times.

8. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and an incorrect address type.

9. Repeat steps 5–6 30 times.

10. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and correct address type.

11. Lower Tester receives an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with an address white listed in the policy applied using correct address type, on the selected advertising channel T_IFS after the end of an advertising packet.

12. Lower Tester receives a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.

13. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.

14. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.

15. Lower Tester address type is set to Random Address type.


17. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from Any (0x01)’.

18. Lower Tester address type is set to Public Address type.


20. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from Any (0x01)’.

21. Lower Tester address type is set to Random Address type.

22. Repeat steps 4–13.

• Expected Outcome

Pass Verdict

The test procedure completes using an interval between the minimum and maximum advertising intervals.

The IUT does not respond to the advertising packets with the black listed addresses.

The IUT does not respond to the advertising packets with the white listed addresses and incorrect address types.
The IUT does respond to the advertising packets with the white listed addresses and correct address types.

The test procedure completes using the filtering policies to black list all unknown devices and to allow scan requests from white listed devices.

- **Notes**
  
  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.10 LL/DDI/ADV/BV-09-C [Connection Request Device Filtering]

- **Test Purpose**
  
  Tests that an advertiser IUT filters initiators according to the white list and filtering policy set.

  The Lower Tester transmits connection requests to the IUT using addresses and address types that either pass or fail the filter, then observes the IUT response on the advertising channels used.

- **Reference**
  
  [3] 4.3.2, 4.4.2.3

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State for the first execution of the test procedure using each channel: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, black list all unknown devices).

  State for the second execution of the test procedure using each channel: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, public address, selected advertising channels, Length of device name used, common device name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, allow scan requests from unknown devices).
• **Test Procedure**

1. **Upper Tester** enables undirected advertising in the **IUT** using public address type, all supported advertising channels and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from White List (0x03)'.

2. **Upper Tester** sends an **HCI_LE_Set_Scan_Response_Data** command with data set to "IUT" and receives an **HCI_Command_Complete** event from the **IUT**.

3. **Lower Tester** address type is set to Public Address type.

4. Configure **Lower Tester** to monitor the advertising and connection procedures of the **IUT** and send a **CONNECT_IND** packet on the selected supported advertising channel (defined as an
IXIT) in response to connectable advertisements. The initiator’s address in the CONNECT_IND PDU shall be formed by using the same address type as the entry on the IUT’s white list but changing the most significant octet of the address to ensure a mis-match.

5. Lower Tester receives an ADV_IND packet from the IUT and responds with a CONNECT_IND packet with the selected address on the selected advertising channel T_IFS after the end of an advertising packet.

6. Lower Tester expects the IUT to continue advertising.

7. Repeat steps 5–6 30 times.

8. Configure Lower Tester to use a device address on the IUT’s white list but an incorrect address type as the address parameter of the CONNECT_IND PDU.

9. Repeat steps 5–6 30 times.

10. Configure Lower Tester to use a device address on the IUT’s white list and correct address type as the address parameter of the CONNECT_IND PDU.

11. Lower Tester receives an ADV_IND packet from the IUT and responds with a CONNECT_IND packet with the selected address white listed in the policy applied, on the selected advertising channel T_IFS after the end of an advertising packet.

12. The Lower Tester receives no ADV_IND packet after advertising interval from the IUT after sending the connection request to indicate that the IUT has stopped advertising. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.

13. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT.

14. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle).

15. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.

16. Lower Tester address type is set to Random Address type.

17. Repeat steps 4–14.

18. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from White List (0x02)’.

19. Lower Tester address type is set to Public Address type.

20. Repeat steps 4–14.

21. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from white list (0x02)’.

22. Lower Tester address type is set to Random Address type.

23. Repeat steps 4–14.

24. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising interval and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)’.

25. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the first supported advertising channel in response to connectable advertisements. The initiator’s address in the CONNECT_IND PDU shall be an address on the IUT’s white list.

26. Lower Tester receives an ADV_IND packet from the IUT and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.

27. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s corrected formatted LL Data Packets on the data channels. If no data packets are received, repeat steps 26 and 27 up to 20 times or until the IUT stops advertising.
28. The Lower Tester receives no ADV_IND packet after advertising interval from the IUT after sending the connection request. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.

29. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 25 and as postamble: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

30. Repeat steps 24–29, except that in step 25, configure Lower Tester to use a device address not on the IUT’s white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT’s white list but changing the most significant octet of the address to ensure a mis-match.

31. Repeat steps 24–29, except that in step 25, configure Lower Tester to use a device address not on the IUT’s white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT’s white list, changing the least significant octet of the address to ensure a mis-match.

32. Repeat steps 24–29, except that in step 25, configure Lower Tester to use a device address not on the IUT’s white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT’s white list, changing both the most and least significant octets of the address to ensure a mis-match.

33. Repeat steps 24–32, except that in step 25, configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the second supported advertising channel in response to connectable advertisements.

34. Repeat steps 24–32, except that in step 25, configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the third supported advertising channel in response to connectable advertisements.

35. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from Any (0x01)’.


- Expected Outcome

Pass Verdict

The test procedure completes using the filtering policies black listing all unknown devices when the policy specifies and the IUT responds to scan requests from unknown devices when allowed.

The IUT reports the connection request packet with a white listed address with an HCI event or with a black listed address with an HCI event, when the policy allows.

4.2.2.11 LL/DDL/ADV/BV-11-C [Directed Advertising Events]

- Test Purpose

Tests that an advertiser IUT sends advertising packets with the directed advertising events’ timing and channel sequence for the maximum time allowed, and accepts a connection request to these packets. The IUT is using High Duty Cycle Connectable Directed Advertising.

The Lower Tester observes the packet and event timing and packet contents from the IUT and requests a connection on the advertising channels used.

- Reference

[3] 4.4.2.4

- Initial Condition

Parameters: LLAdvertiser_Adv_Channel_Map
State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices)).

- **Test Procedure**

  Execute the test procedure using the common connection request packet with the IUT address as the advertiser address.

  ![Diagram](image)

  **Figure 4.50: LL/DDI/ADV/BV-11-C [Directed Advertising Events]**

  1. Configure Lower Tester to start passive scanning.
  2. Upper Tester enables high duty cycle directed advertising in the IUT using all supported advertising channels.
  3. Lower Tester expects the IUT to send ADV_DIRECT_IND packets: A packet starting an event on an applicable advertising channel with the lowest advertising channel index, then optionally
following packets on applicable advertising channels with increasing advertising channel indexes. Expect the intervals between starts of packet on any single channel to be equal to or below 3.75 ms.

4. Repeat until the IUT stops advertising and verify that it stops after 1.28s. For each advertising channel, verify that at least 30 of the intervals between starts of packets on that channel are equal to or below 3.75 ms.

5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT with status parameter set to ‘directed advertising timeout’.

6. Configure Lower Tester to initiate a connection.

7. Upper Tester enables directed advertising in the IUT using all supported advertising channels.

8. Lower Tester receives an ADV_DIRECT_IND packet from the IUT on the selected advertising channel (defined as an IXIT), then responds with a CONNECT_IND packet T_IFS after the end of the advertising packet and does not send any data packets to the IUT.

9. Lower Tester receives no ADV_DIRECT_IND packets from the IUT after the advertising interval.

10. Repeat steps 8–9 until the IUT stops advertising.

11. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 8.

12. Upper Tester receives an HCI_LE_Disconnection_Complete event from the IUT with the reason parameter indicating ‘connection failed to be established’, with the connection handle parameter matching to step 8.

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT stopping advertising to a connection request on each advertising channel.

The number of time intervals measured less than or equal to 3.75 ms between the beginnings of advertising packets on each particular channel is at least 30.

The advertising packets are received over a time period less than or equal to 1.28 s.

The IUT reports the conclusion of advertising with an HCI event.

• Notes

For the total time of advertising, the expression 1.28 s requires a measurement accuracy of 0.013 s. Drift in the total directed advertising time is not significant to this measurement. Note that the total time measured may still be up to 1.293 s (expressed in the measurement accuracy).

4.2.2.12 LL/DDI/ADV/BV-15-C [Discoverable Advertising Events]

• Test Purpose

Tests that an advertiser IUT sends advertising packets of a discoverable undirected event type with correct contents on the selected advertising channel with correct event timing.

The Lower Tester observes the packet and event timing and packet contents on the selected advertising channel.

• Reference

[3] 4.4.2.5

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
State: Discoverable Advertising (selected Adv_interval_Min, selected Adv_interval_Max, supported type of own address, selected advertising channels, length of device name used, common device name)

- Test Procedure

1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester enables discoverable undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising.
3. Lower Tester expects the IUT to send ADV_SCAN_IND packets starting an event on the selected advertising channel.
4. Expect the next event to start after advertising interval time calculated from the start of the first packet.
5. Repeat steps 3–4 until a number of advertising intervals (100) have been detected.
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.
• Expected Outcome

Pass Verdict
The test procedure completes using the selected advertising interval.
The test procedure completes using the selected advertising channel.
The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms.

• Notes
The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.13 LL/DDI/ADV/BV-16-C [Advertising Data: Discoverable]

• Test Purpose
Tests that an advertiser IUT sends advertising packets of discoverable undirected event type with data on the selected advertising channel.
The Upper Tester submits data to the IUT and the Lower Tester observes the IUT including data to the advertising packets on the selected advertising channel.

• Reference
[3] 4.4.2.5

• Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
State: Advertising Parameters Set (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

• Test Procedure
Execute the test procedure using discoverable undirected advertising event type with a selected advertising interval between the minimum and maximum advertising intervals supported using a selected advertising channel, with data lengths of 1 and 31.
Configure Lower Tester to monitor advertising packets from the IUT.

1. Upper Tester enables discoverable undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.

2. Upper Tester sends an `HCI_LE_Set_Advertising_Data` command to the IUT and receives an `HCI_Command_Complete` event in response. The data element used in the command is a number indicating the length of the data. The data length is 1 byte.

3. Upper Tester sends an `HCI_LE_Set_Advertising_Enable` command to the IUT to enable advertising and receives an `HCI_Command_Complete` event in response.
4. Lower Tester expects the IUT to send ADV_SCAN_IND packets including the data submitted in step 3 starting an event on the selected advertising channel.
5. Expect the following event to start after advertising interval time calculating from the start of the first packet.
6. Repeat steps 5–6 until a number of advertising intervals (50) have been detected.
7. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising function and receives an HCI_Command_Complete event in response.
8. Upper Tester sends an HCI_LE_Set_Advertising_Data to configure the IUT to send advertising packets without advertising data and receives an HCI_Command_Complete event in response.
9. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.
10. Lower Tester expects the IUT to send ADV_SCAN_IND packets including no advertising data starting an event on the selected advertising channel.
11. Expect the next event to start after advertising interval time calculating from the start of the first packet.
12. Repeat steps 11–12 until a number of advertising intervals (50) have been detected.
13. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.
14. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and receives an HCI_Command_Complete in response. The data element is a number indicating the length of the data field in the first octet encoded unsigned least significant bit first and the rest of the octets zeroes. The data length is 31 bytes.
15. Repeat steps 4–14.

- Expected Outcome
  
  Pass Verdict
  The test procedure executes with the IUT advertising using the discoverable undirected event type, The IUT transmits data as submitted in the HCI commands.

4.2.2.14 LL/DDI/ADV/BV-17-C [Scan Request: Discoverable]

- Test Purpose
  Tests that an advertiser IUT, advertising with the discoverable undirected event type, responds to a scan request and continues advertising after the response.
  The Lower Tester requests information from the IUT, receives a response, then checks that the advertising resumes.

- Reference
  [3] 4.4.2.5, 4.4.3.2

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.
  
  State for all IUTs Discoverable Advertising (selected Adv_INTERVAL_MIN, supported type of own address, selected advertising channels, length of device name used, selected name) AND White List All Unknown Devices (Allow Scan Request from Any, Allow Connect Request from Any (0x00)).
- **Test Procedure**

  Execute the test procedure using the selected advertising interval.

1. Upper Tester enables discoverable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising.
2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a common public device address as parameter.
3. Configure Scan Response Data in the IUT using device name length of 0 as response data.
4. Lower Tester sends a SCAN_REQ packet on the selected advertising channel after receiving an ADV_SCAN_IND packet from IUT on the advertising channel configured in step 3. The SCAN_REQ is sent T_IFS after the end of an ADV_SCAN_IND packet.
5. Lower Tester receives a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
6. Repeat steps 4–5 30 times.
7. Configure Scan Response Data in the IUT using device name length of 31 as response data.

---

*Figure 4.53: LL/DDI/ADV/BV-17-C [Scan Request: Discoverable]*
8. Repeat steps 4–6.
9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a public device address that differs from the IUT address in the most significant octet as parameter.
10. Configure Scan Response Data in the IUT using device name length of 0 as response data.
11. Repeat steps 4–6.
12. Configure Scan Response Data in the IUT using device name length of 31 as response data.
13. Repeat steps 4–6.
14. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a public device address with address that differs from the IUT address in the most and least significant octets as parameter.
15. Configure Scan Response Data in the IUT using device name length of 0 as response data.
16. Repeat steps 4–6.
17. Configure Scan Response Data in the IUT using device name length of 31 as response data.
18. Repeat steps 4–6.

• Expected Outcome

  Pass Verdict

  The test procedure completes using the selected advertising interval.
  The test procedure completes with the IUT responding on each advertising channel.
  The IUT responds in each case of different scanner address used.
  The timing deviations detected for packets in active mode are within the 2 µs range around T_IFS.

• Notes

  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.15 LL/DDI/ADV/BV-18-C [Device Filtering: Discoverable]

• Test Purpose

  Tests that an advertiser IUT advertising with the discoverable undirected event type filters scanners according to the white list and filtering policy set.
  The Lower Tester transmits scan requests to the IUT using addresses and address types that either pass or fail the filter, then observes the response from the IUT on the advertising channels used.

• Reference

  [3] 4.3.2, 4.4.2.5

• Initial Condition

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State: Discoverable Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, length of device name used, common device name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, black list all unknown devices)
• Test Procedure

1. Upper Tester enables discoverable undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.
2. Lower Tester address type is set to Public Address Type.
3. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send an SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT)
with an address that differs from the IUT address in the least significant octet (an address black listed in the policy applied).

4. Lower Tester receives an ADV_SCAN_IND packet from the IUT and responds with an
   SCAN_REQ packet with the selected address on the selected advertising channel T_IFS after the
   end of an advertising packet.

5. Lower Tester receives no response from the IUT.

6. Repeat steps 4–5 30 times.

7. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and
   send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT)
   with an address white listed in the policy applied and an incorrect address type.

8. Repeat steps 4–6 30 times.

9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and
   send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT)
   with an address white listed in the policy applied and correct address type.

10. Lower Tester receives an ADV_SCAN_IND packet from the IUT and responds with a
    SCAN_REQ packet with an address white listed in the policy applied using correct address type,
    on the selected advertising channel T_IFS after the end of an advertising packet.

11. Lower Tester receives a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS
    after the end of the request packet.

12. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable
    advertising and receives an HCI_Command_Complete event in response.

13. Upper Tester enables discoverable undirected advertising in the IUT using public address type,
    all supported advertising channels, an advertising interval between the minimum and maximum
    advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect
    Request from White List (0x03)’.

14. Lower Tester address type is set to Random Address Type.

15. Repeat steps 3–12.

16. Upper Tester enables discoverable undirected advertising in the IUT using public address type,
    all supported advertising channels, an advertising interval between the minimum and maximum
    advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect
    Request from Any (0x01)’.

17. Lower Tester address type is set to Public Address Type.

18. Repeat steps 3–12.

19. Upper Tester enables discoverable undirected advertising in the IUT using public address type,
    all supported advertising channels, an advertising interval between the minimum and maximum
    advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect
    Request from Any (0x01)’.

20. Lower Tester address type is set to Random Address Type.

21. Repeat steps 3–12.

• Expected Outcome

  Pass Verdict

  The test procedure completes using an interval between the minimum and maximum advertising
  intervals.

  The IUT does not respond to the advertising packets with the black listed addresses.

  The IUT does not respond to the advertising packets with the white listed addresses and incorrect
  address types.

  The IUT does respond to the advertising packets with the white listed addresses and correct address
types.
The test procedure completes using the filtering policies to black list all unknown devices and to allow scan requests from white listed devices.

• Notes

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.16  LL/DDI/ADV/BI-01-C [Scan Request Invalid CRC]

• Test Purpose

Tests that an advertiser IUT ignores a scan request with an invalid checksum and continues advertising.

The Lower Tester sends the invalid scan request and observes the IUT continuing advertising.

• Reference

[3] 3.1, 4.4.2.3

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_INTERVAL_MIN, selected Adv_INTERVAL_MAX, supported type of own address, all advertising channels, Length of device name used, common device name).

• Test Procedure

Execute the test procedure with an advertising interval between the minimum and maximum advertising intervals. For an IUT supporting device filtering, apply the policy of white listing all devices (apply the filtering policy of allowing scan request, connect request from any (0x00).
1. Configure Lower Tester to start an active scanning but sending SCAN_REQ packets with invalid CRC.
2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
3. Lower Tester receives an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with an invalid CRC on the selected advertising channel (defined as an IXIT) T_IFS after the end of an advertising packet.
4. Lower Tester expects the IUT to continue advertising, not responding to the SCAN_REQ packet.
5. Repeat steps 3–4 30 times.
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.

- **Expected Outcome**

  **Pass Verdict**

  The test procedure completes using an interval between the minimum and maximum advertising intervals, using all supported advertising channels.

- **Notes**

  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.
4.2.2.17  LL/DDI/ADV/BI-02-C [Connection Request Invalid CRC]

- **Test Purpose**
  Tests that an advertiser IUT ignores connection requests with an invalid CRC.
  The Lower Tester sends the connection request and observes the IUT continuing advertising.

- **Reference**
  [3] 3.1, 4.4.2.3

- **Initial Condition**
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.
  State: Undirected Advertising (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, supported type of own address, selected advertising channels, Length of device name used, common device name)

- **Test Procedure**
  Execute the test procedure with an advertising interval between the minimum and maximum advertising intervals. Apply the policy of white listing all devices (apply the filtering policy of allowing scan request, connect request from any (0x00)) for an IUT supporting device filtering. The connection request packet contents are defined in 'LL/CON'.

![Diagram of LL/ADV/BI-02-C Test Sequence]

*Figure 4.56: LL/DDI/ADV/BI-02-C [Connection Request Invalid CRC]*

1. Configure Lower Tester to initiate a connection but sending CONNECT_IND packets with invalid CRC.
2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
3. Lower Tester receives an ADV_IND packet from the IUT and responds with a CONNECT_IND packet with an invalid CRC on the selected advertising channel T_IFS after the end of an advertising packet.
4. Lower Tester expects the IUT to continue advertising.
5. Repeat steps 3–4 30 times.
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising and receives an HCI_Command_Complete event in response.

   • Expected Outcome
     
     **Pass Verdict**

     The test procedure completes using an interval between the minimum and maximum advertising intervals on all of the supported advertising channels.

   • Notes
     
     The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.18 LL/DDI/ADV/BV-19-C [Low Duty Cycle Directed Advertising Events]

   • Test Purpose

     Test that an advertiser IUT sends advertising packets of the low duty cycle directed advertising event type with correct contents on a selected advertising channel with correct event timing, and accepts a connection request to these packets.

     The Lower Tester observes the packet and event timing and packet contents on the selected advertising channel. The Lower Tester also solicits a connection on the selected advertising channel.

   • Reference

     [3] 4.4.2.4

   • Initial Condition

     Parameters: LL_advertiser_advInterval_Min, LL_advertiser_advInterval_Max, LL_advertiser_Adv_Channel_Map

     State: Low Duty Cycle Directed Advertising (selected AdvInterval_Min, selected AdvInterval_Max, supported type of own address, public initiator address, Lower Tester address, selected advertising channels)
• Test Procedure

Figure 4.57: LL/DDI/ADV/BV-19-C [Low Duty Cycle Directed Advertising Events]

1. Configure Lower Tester to start scanning and monitor advertising packets from the IUT.
2. Upper Tester enables low duty cycle directed advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising.
3. Lower Tester expects the IUT to send ADV_DIRECT_IND packets starting an event on the selected advertising channel.
4. Expect the next event to start after the advertising interval time calculated from the start of the first packet.
5. Repeat steps 3–4 until the number of advertising intervals (100) have been detected.
6. Configure the Lower Tester to initiate a connection.
7. Lower Tester receives an ADV_DIRECT_IND packet from the IUT on the selected advertising channel (defined as an IXIT), then responds with a CONNECT_IND packet T_IFS after the end of the advertising packet and does not send any data packets to the IUT.
8. Lower Tester receives no ADV_DIRECT_IND packets from the IUT after the advertising interval.
9. Repeat steps 7–8 until the IUT stops advertising.
10. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 7.
11. Upper Tester receives an HCI_Disconnection_Complete event from the IUT once the Establishment Timeout has expired.

- Expected Outcome
  
  **Pass Verdict**
  
  The test procedure completes using the selected advertising interval.
  
  The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms.
  
  The test procedure completes with the IUT stopping advertising after receiving a connection request on the selected advertising channel.
  
  The IUT reports the conclusion of advertising with an HCI event.

- Notes
  
  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and the test result criteria.

### 4.2.2.19 LL/DDI/ADV/BV-20-C [Advertising Always Using the LE 1M PHY]

- Test Purpose
  
  Test that an advertiser IUT sends advertising packets of an event with correct contents on all applicable advertising channels using the LE 1M PHY, even when the host has indicated that it prefers the LE 2M PHY.
  
  The Lower Tester observes the packet and event timing and packet contents on the advertising channels used and confirms they can be received using the LE 1M PHY.

- Reference
  
  [10] 2.3

- Initial Condition
  
  Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map
  
  State: Undirected Advertising (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_MAX, supported type of own address, selected advertising channels, length of device name used, common device name)
Test Procedure

1. Configure Lower Tester to monitor advertising packets from the IUT. Lower Tester will only accept advertising packets sent using the LE 1M PHY setting. Lower Tester will scan for at least 30 advertising intervals on each advertising channel (for example, scan on channel 37 for the first 30 intervals, then on channel 38 for another 30 intervals, then finally on channel 39 for the last 30 intervals).

2. Upper Tester sends a LE_Set_Default_PHY command to the IUT, with the ALL_PHYS field set to zero, and the TX_PHYS and RX_PHYS fields both set to prefer the LE 2M PHY.

3. Upper Tester enables undirected advertising in the IUT using all supported advertising channels and minimum advertising interval.

4. Lower Tester expects the IUT to send ADV_IND packets starting an event on an applicable advertising channel using the LE 1M PHY.

5. Repeat step 4 until at least 90 advertising packets have been detected, i.e., at least 30 packets on each channel.

Figure 4.58: LL/DDI/ADV/BV-20-C [Advertising Always Using the LE 1M PHY]
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

- Expected Outcome
  
  Pass Verdict
  
  All advertising events are transmitted using the LE 1M PHY and are received properly by the Lower Tester using the LE 1M PHY.

4.2.2.20 LL/DDI/ADV/BV-21-C [Extended Advertising, Legacy PDUs, Non-Connectable]

- Test Purpose
  
  Tests that an advertiser IUT sends advertising packets of a non-connectable event type with data on a selected primary advertising channel using legacy PDU types and extended advertising HCI commands.
  
  The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the selected primary advertising channel.

- Reference
  
  [10] 4.4.2.6

- Initial Condition
  
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  
  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- Test Procedure
  
  Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using a selected primary advertising channel, with data lengths of 1, 0, and 31.
For each round from 1 to 3 based on Table 4.2:

1. Configure Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using a selected primary advertising channel and minimum advertising interval. The Advertising_Event_Properties parameter shall be set to 00010000b (ADV_NONCONN_IND legacy PDU).

3. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command to the IUT with values according to Table 4.2 and receives an HCI_Command_Complete in response.
4. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to the IUT to enable advertising and receives an HCI_Command_Complete event in response.
5. Lower Tester expects the IUT to send ADV_NONCONN_IND packets including the data submitted in step 3 starting an event on the selected primary advertising channel.
6. Expect the following event to start after advertising interval time calculating from the start of the first packet.
7. Repeat steps 5–6 until a number of advertising intervals (50) have been detected.
8. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to the IUT to disable advertising function and receives an HCI_Command_Complete event in response.
9. Repeat steps 3–8 for each Round shown in Table 4.2.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table 4.2: Payload contents for each case variation.

<table>
<thead>
<tr>
<th>Round</th>
<th>Data Length</th>
<th>Data Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>No data</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>0xF8, [0x00]…</td>
</tr>
</tbody>
</table>

- **Expected Outcome**
  - **Pass Verdict**
    - The test procedure executes with the IUT advertising using non-connectable event type. The ADV_NONCONN_IND PDU is utilized.
    - The IUT transmits data as submitted in the HCI commands.

4.2.2.21 **Extended Advertising, Legacy PDUs, Undirected**

- **Test Purpose**
  - Tests that an advertiser IUT sends advertising packets of an undirected type of event with data on all advertising channels using legacy PDU types and extended advertising HCI commands.
  - The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used. The Lower Tester confirms the IUT sets ChSel as specified in Table 4.4 in the legacy PDU.

- **Reference**
  - [10] 4.4.2.3

- **Initial Condition**
  - Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  - State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- **Test Procedure**
  - Execute the test procedure using undirected advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels, with data lengths of 1, 0, and 31.
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. The Advertising_Event_Properties parameter shall be set to 00010011b (ADV_IND legacy PDU).

For each round from 1 to 3 based on Table 4.3:

3. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command to the IUT with values according to Table 4.3 and receives an HCI_Command_Complete in response.
4. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to the IUT to enable advertising with Duration[0] set to 0x0000 (continue advertising until disabled), and receives an HCI_Command_Complete event in response.
5. Lower Tester scans on a single primary advertising channel as indicated in Table 4.3 and expects the IUT to send ADV_IND packets, with ChSel set as specified in Table 4.4, including the data submitted in step 3 starting an event on the applicable primary advertising channel.
6. Repeat step 5 until a number of advertising intervals (50) have been detected.
7. Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to the IUT to disable advertising function and receives an HCI_Command_Complete event in response.
8. Repeat steps 3–7 for each Round shown in Table 4.3.
Link Layer (LL) / Test Suite

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Length</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 4.3: Payload contents for each case variation.

- Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>ChSel</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2.21.1</td>
<td>1</td>
</tr>
<tr>
<td>4.2.2.21.2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.4: Extended Advertising, Legacy PDUs, Undirected Test Cases

- Expected Outcome

Pass Verdict

The test procedure executes with the IUT advertising using Connectable Undirected event type. The ADV_IND PDU is utilized, with ChSel set as specified in Table 4.4.

The IUT transmits data as submitted in the HCI commands.

4.2.2.22 Extended Advertising, Non-Connectable

- Test Purpose

Tests that an advertiser IUT sends non-connectable ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with and without data, along with chaining, are tested. Undirected and Directed events are tested.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.

- Reference

[10] 4.4.2.6, 4.4.2.10

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_MAX, selected type of advertising events, supported type of own address, selected advertising channel map)
• Test Procedure

The following test procedure applies to the test cases listed in Table 4.6, the only change is the variation of the PHYs used.

Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.

1. The Upper Tester sends an HCI_LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.
For each round based on Table 4.5:

2. If the Data Length listed in Table 4.5 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 14.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to the value specified in Table 4.5 for this round. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set to the values specified in Table 4.6. If the Advertising_Event_Properties value for this Round specifies directed advertising, the Peer_Address_Type shall be set to 0x00 (Public Device Address), and the Peer_Address shall be set to the Lower Tester’s address.

4. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT with values according to Table 4.5 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter shall be set to the value specified in Table 4.5 for this round. The Max_Extended_Advertising_Events[0] parameter shall be set to the value specified in Table 4.5 for this round.

6. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 00b. The ADV_EXT_IND PDU shall not include the SuppInfo, SyncInfo, TxPower, ACAD, or AdvData fields. If advertising data was set in step 4, the ADV_EXT_IND PDU shall include the AuxPtr field; otherwise, the ADV_EXT_IND PDU may include the AuxPtr field. If the AuxPtr field is included, the ADV_EXT_IND PDU shall also include the ADI field with the SID set to the value used in step 3; otherwise that field shall not be included.

7. If the AuxPtr is absent, skip to step 10.

8. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b. The AUX_ADV_IND PDU shall not include the SuppInfo, SyncInfo, or TxPower fields. The AUX_ADV_IND PDU shall include the ADI field matching the ADI field from step 6. If the AUX_ADV_IND PDU does not contain all the data submitted in step 4 (if any), it shall include an AuxPtr field.

9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall include the ADI field matching the ADI field from step 6 and the AdvData field containing additional data submitted in step 4. The AUX_CHAIN_IND PDU shall not include the AdvA, TargetA, SuppInfo, TxPower, or SyncInfo fields. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.

10. If the Max_Extended_Advertising_Events was set to a value different than 0, repeat steps 6–9 until the IUT stops advertising. Afterwards, the Lower Tester confirms that the IUT did not send more than Max_Extended_Advertising_Events advertising events. Upper Tester shall receive LE Advertising Set Terminated event with ErrorCode 0x43. Skip to step 13.

11. Otherwise if Duration was set to a value different than 0, repeat steps 6–9 until the amount of time specified for Duration has elapsed. Afterwards, the Lower Tester confirms that the IUT does not start any additional advertising events. Upper Tester shall receive LE Advertising Set Terminated event with ErrorCode 0x3C. Skip to step 13.

12. Otherwise, repeat steps 6–9 until a number of advertising intervals (10) have been detected.
13. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. Repeat steps 2–13 for each Round shown in Table 4.5.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 4)</th>
<th>HCI_LE_Set_Extended_Advertising_Enable (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Data Length</td>
<td>Fragment_Preference</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>2</td>
<td>0x0000</td>
<td>31</td>
<td>0x00</td>
</tr>
<tr>
<td>3</td>
<td>0x0000</td>
<td>474</td>
<td>0x00</td>
</tr>
<tr>
<td>4</td>
<td>0x0000</td>
<td>711</td>
<td>0x00</td>
</tr>
<tr>
<td>5</td>
<td>0x0000</td>
<td>948</td>
<td>0x00</td>
</tr>
<tr>
<td>6</td>
<td>0x0000</td>
<td>Maximum_Advertising_Data_Length</td>
<td>0x00</td>
</tr>
<tr>
<td>7</td>
<td>0x0000</td>
<td>Maximum_Advertising_Data_Length</td>
<td>0x01</td>
</tr>
<tr>
<td>8</td>
<td>0x0004</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>9</td>
<td>0x0004</td>
<td>251</td>
<td>0x00</td>
</tr>
<tr>
<td>10</td>
<td>0x0004</td>
<td>Maximum_Advertising_Data_Length</td>
<td>0x00</td>
</tr>
<tr>
<td>11</td>
<td>0x0000</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>12</td>
<td>0x0004</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>13</td>
<td>0x0000</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>14</td>
<td>0x0004</td>
<td>0</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Table 4.5: Payload contents for each case variation
• Expected Outcome

The following test procedure applies to the test cases listed in Table 4.6, the only change is the variation of the PHYs used.

Pass Verdict

The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND PDU on the primary advertising channel.
- If advertising data was specified for the round, the ADV_EXT_IND PDU includes the AuxPtr field referring to an AUX_ADV_IND PDU on the secondary channel.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- If the advertising was directed, the TargetA field containing the Lower Tester’s address specified in the HCI_LE_Set_Extended_Advertising_Parameters command is included in either the ADV_EXT_IND PDU or the AUX_ADV_IND PDU, but not both.
- [LL/DDI/ADV/BV-48-C only] If the advertising was directed and the ADV_EXT_IND PDU contained an AuxPtr, the TargetA field is included in the AUX_ADV_IND PDU.
- The SuppInfo, SyncInfo, TxPower, ACAD, and AdvData fields are not included in the ADV_EXT_IND PDU.
- The SuppInfo, SyncInfo, and TxPower fields are not included in the AUX_ADV_IND PDU.
- If the ADV_EXT_IND PDU includes the AuxPtr field, the ADV_EXT_IND and AUX_ADV_IND PDUs and any AUX_CHAIN_IND PDUs contain the ADI field with the SID set to the value specified in the HCI_LE_Set_Extended_Advertising_Parameters command and the same DID value.
- If advertising data was specified for the round, the Lower Tester receives all the data sent in step 4, and no further data, in the AUX_ADV_IND PDU and zero or more AUX_CHAIN_IND PDUs. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs shall not contain any data.
- If one or more AUX_CHAIN_IND PDUs were sent, each includes the ADI field matching the ADI field included in the AUX_ADV_IND PDU, and none of them include the AdvA, TargetA, SuppInfo, TxPower, or SyncInfo fields.
- When the Duration parameter is set, the IUT does not start any new advertising events after the time specified for Duration has elapsed. An HCI LE Advertising Set Terminated event shall be received with the correct error code.
- When the Max_Extended_Advertising_Events parameter is set, the IUT does not start more than the Max_Extended_Advertising_Events. An HCI LE Advertising Set Terminated event shall be received with correct error code.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs used in step 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary_Advertising_PHY</td>
</tr>
<tr>
<td>4.2.2.22.1_LL/DDI/ADV/BV-47-C Extended Advertising, Non-Connectable</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>Test Case</td>
<td>PHYs used in step 3</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>[Extended Advertising, Non-Connectable – LE Coded PHY]</td>
<td></td>
</tr>
<tr>
<td>4.2.2.22.2 LL/DDI/ADV/BV-48-C</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
<tr>
<td>[Extended Advertising, Non-Connectable – LE 2M PHY]</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
<tr>
<td>4.2.2.22.3 LL/DDI/ADV/BV-49-C</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>[Extended Advertising, Non-Connectable – LE 2M PHY]</td>
<td>0x02 (LE 2M PHY)</td>
</tr>
</tbody>
</table>

Table 4.6: Extended Advertising, Non-Connectable test cases

- The minimum actual advertising duration required is the duration specified plus negative drift plus negative jitter: (10 ms * Duration) - (10 ms * Duration * 500 ppm) - 0.016 ms.
- The maximum actual duration is (10 ms * Duration + 10.000 ms) * (1 + 500 ppm) + 0.016 ms.
- The notes in LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events] describe the reasoning of the timing measurements and test result criteria.

4.2.2.23 Extended Advertising, Scannable

- Test Purpose
  Tests that an advertiser IUT sends scannable ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel with the correct payload fields, timing, and channel sequence for the maximum time allowed. Tests that an advertiser IUT responds to a scan request on the secondary channel and continues advertising after the response. Scan response data chaining is tested. Undirected and Directed events are tested.
  The Lower Tester requests information from the IUT, receives a response, then checks that the advertising resumes.

- Reference
  [10] 4.4.2.5.2, 4.4.2.8,
  [13] 2.3.2.3

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.
  State: Scannable Advertising (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, supported type of own address, selected advertising channels, length of device name used, selected name).

- Test Procedure
  Execute the test procedure with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.
1. The Upper Tester sends an `HCI_LE_Read_Maximum_Advertising_Data_Length` command to the IUT and expects the IUT to return a `Maximum_Advertising_Data_Length` between 0x001F and 0x0672. The Upper Tester stores the `Maximum_Advertising_Data_Length` for future use.

For each round from 1 to 11 based on Table 4.7:

2. If the Data Length listed in Table 4.7 for the current Round is less than or equal to the `Maximum_Advertising_Data_Length` proceed to step 3, otherwise skip to step 13.

3. The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. `Advertising_Event_Properties` and `Scan_Response_Notification_Enable` parameters shall be set to the value specified in Table 4.7.
for this round. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set as specified in Table 4.7. If the Advertising_Event_Properties value for this Round specifies directed advertising, the Peer_Address_Type shall be set to 0x00 (Public Device Address), and the Peer_Address shall be set to the Lower Tester’s address.

4. The Upper Tester sends one or more HCI_LE_Set_Extended.Scan_Response_Data commands to the IUT with values according to Table 4.7 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

6. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 10b with the AuxPtr Extended Header field present. The ADV_EXT_IND PDU shall include the ADI field with the SID set to the value used in step 3. The ADV_EXT_IND PDU shall not include the SuppInfo, SyncInfo, TxPower, ACAD, or TxPower fields.

7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 10b. The AUX_ADV_IND PDU shall include the ADI field matching the ADI field from step 6. The AUX_ADV_IND PDU shall not include the SuppInfo, AuxPtr, SyncInfo, TxPower, or AdvData fields.

8. The Lower Tester responds with an AUX_SCAN_REQ PDU T_IFS after the end of the AUX_ADV_IND PDU on the secondary advertising channel with ScanA set to the Lower Tester’s address and AdvA as shown in Table 4.7.

9. If the AUX_SCAN_REQ PDU has an AdvA not equal to the IUT’s address, the Lower Tester receives no AUX_SCAN_RSP packet from the IUT. Skip to step 12.

10. The Lower Tester receives an AUX_SCAN_RSP packet from the IUT T_IFS after the end of the AUX_SCAN_REQ PDU with AdvMode set to 00b, AdvA set to the IUT’s advertising address from step 3, TargetA and SuppInfo not present, and ADI as specified in Table 4.8. If the AUX_SCAN_RSP PDU does not contain all the data submitted in step 4 (if any), it shall include an AuxPtr field.

11. If the AUX_SCAN_RSP PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall include the AdvData field containing additional data submitted in step 4. The AUX_CHAIN_IND PDU shall not include the AdvA, TargetA, SuppInfo, TxPower, or SyncInfo fields. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.

12. If the Scan_Response_Notification_Parameter value specified for step 3 in this round was 0x01 (notifications enabled), the Upper Tester receives an HCI_LE_Scan_Request_Received event from the IUT with the advertising handle used in steps 3–5 and the Lower Tester’s address from step 8.

13. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

14. Repeat steps 2–13 for each Round shown in Table 4.7.
<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3)</th>
<th>HCI_LE_Set_Extended_Scan_Response_Data (Step 4)</th>
<th>AUX_SCAN_REQ PDU (Step 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Scan_Response_Notification_Enable</td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0x0002</td>
<td>0x00</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0x0002</td>
<td>0x00</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>0x0002</td>
<td>0x00</td>
<td>474</td>
</tr>
<tr>
<td>4</td>
<td>0x0002</td>
<td>0x00</td>
<td>711</td>
</tr>
<tr>
<td>5</td>
<td>0x0002</td>
<td>0x00</td>
<td>948</td>
</tr>
<tr>
<td>6</td>
<td>0x0002</td>
<td>0x00</td>
<td>Maximum_Advertising_Data_Length</td>
</tr>
<tr>
<td>7</td>
<td>0x0002</td>
<td>0x01</td>
<td>Maximum_Advertising_Data_Length</td>
</tr>
<tr>
<td>8</td>
<td>0x0002</td>
<td>0x00</td>
<td>31</td>
</tr>
<tr>
<td>9</td>
<td>0x0006</td>
<td>0x00</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0x0006</td>
<td>0x00</td>
<td>251</td>
</tr>
<tr>
<td>11</td>
<td>0x0006</td>
<td>0x00</td>
<td>Maximum_Advertising_Data_Length</td>
</tr>
<tr>
<td>12</td>
<td>0x0006</td>
<td>0x00</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 4.7: Payload contents for each case variation

- Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>ADI Field</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Advertising PHY</td>
<td>Secondary Advertising PHY</td>
</tr>
<tr>
<td><strong>4.2.2.23.1 LL/DDI/ADV/BV-25-C [Extended Advertising, Scannable – ADI not allowed in scan response]</strong></td>
<td>Not Present</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
</tbody>
</table>
### Test Case | ADI Field | Primary Advertising PHY | Secondary Advertising PHY
--- | --- | --- | ---
4.2.2.23.2 | LL/DDI/ADV/BV-45-C [Extended Advertising, Scannable – ADI allowed in scan response] | Not Present or ADI Field in AUX_ADV_IND PDU | 0x01 (LE 1M PHY) | 0x01 (LE 1M PHY)
4.2.2.23.3 | LL/DDI/ADV/BV-51-C [Extended Advertising, Scannable – ADI not allowed in scan response – LE 2M PHY] | Not Present | 0x01 (LE 1M PHY) | 0x02 (LE 2M PHY)
4.2.2.23.4 | LL/DDI/ADV/BV-52-C [Extended Advertising, Scannable – ADI allowed in scan response – LE 2M PHY] | Not Present or ADI Field in AUX_ADV_IND PDU | 0x01 (LE 1M PHY) | 0x02 (LE 2M PHY)
4.2.2.23.5 | LL/DDI/ADV/BV-53-C [Extended Advertising, Scannable – ADI not allowed in scan response – LE Coded PHY] | Not Present | 0x03 (LE Coded PHY) | 0x03 (LE Coded PHY)
4.2.2.23.6 | LL/DDI/ADV/BV-54-C [Extended Advertising, Scannable – ADI allowed in scan response – LE Coded PHY] | Not Present or ADI Field in AUX_ADV_IND PDU | 0x03 (LE Coded PHY) | 0x03 (LE Coded PHY)

Table 4.8: Extended Advertising, Scannable Test Cases

- **Expected Outcome**

  **Pass Verdict**

  The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

  For all rounds described in the test procedure, the following condition shall occur:

  - The IUT sends an ADV_EXT_IND PDU on the primary advertising channel with an AuxPtr field referring to an AUX_ADV_IND PDU on the secondary advertising channel.
  - The IUT responds to the AUX_SCAN_REQ within the 2 \( \mu \)s range around T_IFS.
  - The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
  - The ADV_EXT_IND PDU does not include the SuppInfo, SyncInfo, TxPower, ACAD, or TxPower fields.
  - The AUX_ADV_IND PDU does not include the SuppInfo, AuxPtr, SyncInfo, TxPower, or AdvData fields.
  - The ADV_EXT_IND and AUX_ADV_IND PDUs both contain the ADI field with the SID set to the value specified in the HCI_LE_Set_Extended_Advertising_Parameters command.
- If scan response data was specified for the round, the Lower Tester receives all the data sent in step 4, and no further data, in the AUX_SCAN_RSP PDU and zero or more AUX_CHAIN_IND PDUs. If no data was specified, the AUX_SCAN_RSP PDU and any AUX_CHAIN_IND PDUs shall not contain any data.

- If the scan request notifications were enabled for the round, the Upper Tester receives an HCI_LE_Scan_Request_Received event from the IUT with the advertising handle and the Lower Tester’s address.

- If the AUX_SCAN_REQ PDU has an AdvA not equal to the IUT’s address then the IUT shall not respond.

- The AUX_SCAN_RSP PDU includes or excludes the fields as specified in step 10.

- If one or more AUX_CHAIN_IND PDUs were sent, none of them include the AdvA, TargetA, SuppInfo, ADI, TxPower, or SyncInfo fields.

**Notes**

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

### 4.2.2.24 Extended Advertising, Periodic Advertising

**Test Purpose**

Tests that an advertiser IUT sends ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU containing a SyncInfo field that further refers to AUX_SYNC_IND PDUs on the secondary advertising channel with the correct payload fields, timing, and channel sequence. Advertisements with and without data, along with chaining, are tested. The Lower Tester confirms that Channel Selection Algorithm #2 is utilized for the periodic advertisements.

The Upper Tester submits data of varying lengths to the IUT for periodic advertising, and the Lower Tester observes the IUT performing periodic advertising of the data.

**Reference**

[10] 4.4.2.13.1

**Initial Condition**

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv_INTERVAL_MIN, selected Adv_INTERVAL_MAX, selected type of advertising events, supported type of own address, selected advertising channel map)

**Test Procedure**

Execute the test procedure using periodic advertising events with a selected periodic advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels. Confirm that the IUT continues periodic advertising when extended advertising is disabled.
For each round from 1 to 6

**HCI LE Set Extended Advertising Parameters**
(Extended Advertising)
**HCI Command Complete Event**
(Status: 0x00)

**HCI LE Set Periodic Advertising Parameters**
**HCI Command Complete Event**
(Status: 0x00)

**HCI LE Read Maximum Advertising Data Length**
**HCI Command Complete Event**
(Status: 0x00, Max_Adv_Data_Length)

Repeat until all data is set

**HCI LE Set Periodic Advertising Data**
**HCI Command Complete Event**
(Status: 0x00)

**HCI LE Set Periodic Advertising Enable**
(Enable)
**HCI Command Complete Event**
(Status: 0x00)

**HCI LE Set Extended Advertising Enable**
(Enable)
**HCI Command Complete Event**
(Status: 0x00)

*Figure 4.63: Extended Advertising, Periodic Advertising – Part A*
1. The Upper Tester sends an HCI_LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x00 and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

For each round from 1 to 6 based on Table 4.9:

2. If the Data Length listed in Table 4.9 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 17.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The
Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set to the values specified in Table 4.10.

4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and selected periodic interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.

5. The Upper Tester sends one or more HCI_LE_Set_Periodic_Advertising_Data commands to the IUT with values according to Table 4.9 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 252 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

6. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (Periodic Advertising).

7. The Upper Tester enables advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

8. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present.

9. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields present.

10. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the secondary advertising channel using the index selected by the LE Channel Selection Algorithm #2 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall have the AdvMode field set to 00b. If the AUX_SYNC_IND PDU does not contain all the data submitted in step 5 (if any), it shall include an AuxPtr field.

11. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.

12. Repeat steps 8–11 100 times.

13. The Upper Tester disables extended advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command but maintains synchronization with the IUT’s periodic advertising.

14. The Lower Tester confirms that periodic advertising continues when extended advertising is disabled by repeating steps 10–11 100 times.

15. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

16. The Upper Tester clears the advertising configuration using the HCI_LE_Clear_Advertising_Sets command.

17. Repeat steps 2–16 for each Round shown in Table 4.9.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Periodic_Advertising_Data (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>252</td>
</tr>
<tr>
<td>Round</td>
<td>HCI_LE_Set_Periodic_Advertising_Data (Step 5)</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Data Length</td>
</tr>
<tr>
<td>3</td>
<td>474</td>
</tr>
<tr>
<td>4</td>
<td>711</td>
</tr>
<tr>
<td>5</td>
<td>948</td>
</tr>
<tr>
<td>6</td>
<td>Maximum_Advertising_Data_Length</td>
</tr>
</tbody>
</table>

Table 4.9: Payload contents for each case variation.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Advertising PHY</td>
</tr>
<tr>
<td>4.2.2.24.1 LL/DDI/ADV/BV-26-C [Extended Advertising, Periodic Advertising – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.2.2.24.2 LL/DDI/ADV/BV-55-C [Extended Advertising, Periodic Advertising – LE 2M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.2.2.24.3 LL/DDI/ADV/BV-56-C [Extended Advertising, Periodic Advertising – LE Coded PHY]</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

Table 4.10: Extended Advertising, Periodic Advertising Test Cases

• Expected Outcome

Pass Verdict

The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- The IUT sends AUX_SYNC_IND PDUs on the secondary advertising channel using indices selected by the Channel Selection Algorithm #2.
- If periodic advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_SYNC_IND PDU and zero or more AUX_CHAIN_IND PDUs. If no data was specified, any AUX_SYNC_IND and AUX_CHAIN_IND PDUs shall not contain any data.

**Notes**

- The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.
- The periodic advertising interval used in the test needs to be large enough to allow the Controller to transmit all the data in each of the advertising sets.

### 4.2.2.25 LL/DDI/ADV/BV-27-C [Extended Advertising, Host Modifying Data and ADI]

**Test Purpose**

Tests that an advertiser IUT sends non-connectable undirected advertising packets with the ADV_EXT_IND PDU on the primary advertising channel with the correct payload fields, timing, and channel sequence for the maximum time allowed. Upper Tester modifies the data to be advertised each round and the Lower Tester confirms the data is modified. The ADI field is present and the Lower Tester confirms the Data ID changes if the data changes. Data chaining is also tested.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.

**Reference**

[10] 4.4.2.6, 2.3.4.4

**Initial Condition**

Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_A dv_Channel_Map

State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

**Test Procedure**

Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.
1. The Upper Tester sends a LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

For each round from 1 to 3 based on Table 4.11:

3. If the Data Length listed in Table 4.11 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 4, otherwise skip to step 10.

4. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT with values according to Table 4.11 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one
Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

5. On Round 1 only the Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

6. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present. The ADI field shall be present and contain the Advertising Set ID (SID) used by the Upper Tester in step 3 and an Advertising Data ID.

7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set to 00b and an ADI field matching the ADI field of the ADV_EXT_IND in step 6. If the AUX_ADV_IND PDU does not contain all the data submitted in step 4, it shall include an AuxPtr field.

8. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 4. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.

9. Except for the first advertisement in round 1, the Lower Tester compares the data in the AUX_ADV_IND and any AUX_CHAIN_IND PDUs with that from the previous advertisement (the data shall be concatenated together and the boundaries between PDUs ignored). If the data is not the same but the Advertising Data ID field has not changed, a Fail Verdict is recorded.

10. Repeat steps 6–9 10 times.

11. Repeat steps 3–10 for each Round shown in Table 4.11.

12. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>Maximum_Advertising_Data_Length</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>251</td>
</tr>
</tbody>
</table>

*Table 4.11: Payload contents for each case variation.*

- **Expected Outcome**

**Pass Verdict**

The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel. The ADI field shall be present and contain the Advertising Set ID (SID) used by the Upper Tester in step 2 and an Advertising Data ID.

- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- The Lower Tester receives all the data sent in step 4, and no further data, in the AUX_ADV_IND PDU and zero or more AUX_CHAIN_IND PDUs when only reading data from PDUs with a new DID. The DID shall change every time the data advertised changes.

- The Advertising Data ID changes whenever the data changes.

• Notes

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.26 LL/DDI/ADV/BV-28-C [Extended Advertising, Overlapping Extended Advertising Events]

• Test Purpose

Tests that an advertiser IUT sends ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel. Proper handling of the Secondary_Advertising_Max_Skip parameter is tested.

The Lower Tester observes the event timing and packet contents on the advertising channels in use.

• Reference

[10] 4.4.2.2.2, 2.3.4.5

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)
• **Test Procedure**

For each round from 1 to 6 based on Table 4.12:

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties shall be set according to Table 4.12. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M). Secondary_Advertising_Max_Skip shall be set according to Table 4.12.

2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command to the IUT with length 1 and using a random octet from 1 to 255 as the payload.

3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

4. The Lower Tester receives advertising events consisting of ADV_EXT_IND packets from the IUT with the AuxPtr Extended Header field present and the AdvMode set according to expected properties in Table 4.12.

5. The Lower Tester utilizes the AuxPtr’s Aux Offset and Offset Units to calculate the expected time when the IUT will send an AUX_ADV_IND PDU on the secondary advertising channel.

6. Repeat steps 4–5 until the target time from step 5 is reached, recalculating the target time each round, expecting all target times to be within one Offset Unit of each other.

*Figure 4.66: LL/DDI/ADV/BV-28-C [Extended Advertising, Overlapping Extended Advertising Events]*
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in Table 4.12.

8. The Lower Tester records the total number of advertising events in which the IUT skipped sending an AUX_ADV_IND PDU.

9. Repeat steps 4–8 the number of times specified in Table 4.12.

10. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

11. Repeat steps 1–10 for each Round shown in Table 4.12.

<table>
<thead>
<tr>
<th>Round</th>
<th>Advertising_Event_Properties (Step 1)</th>
<th>Secondary_Advertising_Max_Skip (Step 1)</th>
<th>Repeat count (Step 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x01</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0x0000</td>
<td>0x0F</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>0x0000</td>
<td>0xFF</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>0x0001</td>
<td>0x08</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>0x0005</td>
<td>0x08</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>0x0005</td>
<td>0x08</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 4.12: Advertising properties for each case variation.

- Expected Outcome
  
  Pass Verdict
  
  For all rounds described in the test procedure, the following condition shall occur:
  
  - The IUT sends ADV_EXT_IND PDUs with the AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
  
  - The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
  
  - The AuxPtrs in each ADV_EXT_IND PDU sent in overlapping extended advertising events have Aux Offset and Offset Units values that refer to the same time within one Offset Unit.
  
  - The total number of advertising events in which the IUT skipped sending an AUX_ADV_IND PDU is less than or equal to Secondary_Advertising_Max_Skip. No more than Secondary_Advertising_Max_Skip+1 extended advertising events overlap.

- Notes
  
  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.
4.2.2.27 LL/DDI/ADV/BV-29-C [Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY]

- **Test Purpose**
  Tests that an advertiser IUT can support multiple advertising sets using the LE 1M PHY with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with and without data are tested.
  The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.

- **Reference**
  [10] 4.4.2.6

- **Initial Condition**
  Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map
  State: Advertising Parameters Set (selected Adv_Min, selected Adv_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- **Test Procedure**
  Execute the test procedure using multiple types of advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.

**Figure 4.67: LL/DDI/ADV/BV-29-C [Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY] – Part A**
1. The Upper Tester sends a LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

For each round from 1 to 7 based on Table 4.13:

2. If the Data Length listed in Table 4.13 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 12.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in Table 4.13. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter
shall be set according to the Second Set column in Table 4.13. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to Table 4.13 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).

7. The Lower Tester receives interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in Table 4.13 for each set.

8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in Table 4.13 for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in step 5 for the set (if any), it shall include an AuxPtr field.

9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.

10. Repeat steps 7–9 10 times.

11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.

12. The Upper Tester clears the advertising sets using one HCI_LE_Clear_Advertising_Sets command.

13. Repeat steps 2–12 for each Round shown in Table 4.13.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3)</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 4)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Advertising_Event_Properties</td>
<td>Data Length</td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0x0001</td>
<td>0x0000</td>
<td>0</td>
<td>191</td>
</tr>
</tbody>
</table>
Table 4.13: Payload contents for each case variation

<table>
<thead>
<tr>
<th>Round</th>
<th>Advertising Event Properties</th>
<th>Advertising Event Properties</th>
<th>Data Length</th>
<th>Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0x0000</td>
<td>0x0005</td>
<td>191</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0x0000</td>
<td>0x0000</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>5</td>
<td>0x0000</td>
<td>0x0004</td>
<td>251</td>
<td>251</td>
</tr>
<tr>
<td>6</td>
<td>0x0000</td>
<td>0x0001</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0x0000</td>
<td>0x0000</td>
<td>474</td>
<td>0</td>
</tr>
</tbody>
</table>

- Expected Outcome
  
  **Pass Verdict**
  
The IUT returns a Maximum_Advertising_Data Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT interlaces multiple advertising events for each set. Each set consists of ADV_EXT_IND PDUs with an optional AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- For each set, if advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs shall not contain any data.

- Notes
  
The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.
4.2.2.28 LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY]

- **Test Purpose**
  Tests that an advertiser IUT can support multiple advertising sets using the LE Coded PHY with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with and without data are tested.

  The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.

- **Reference**
  [10] 4.4.2.6

- **Initial Condition**
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- **Test Procedure**
  Execute the test procedure using multiple types of advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.

  ![Diagram](image)

  **Figure 4.69:** LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY] – Part A
Figure 4.70: LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY] – Part B

1. The Upper Tester sends a LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

For each round from 1–7 Based on Table 4.14:

2. If the Data Length listed in Table 4.14 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 12.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in Table 4.14. The Own_ADDRESS_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in Table 4.14. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).

5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to Table 4.14 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).

7. The Lower Tester receives interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in Table 4.14 for each set using the LE Coded PHY.

8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the second advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in Table 4.14 for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in step 5 for the set (if any), it shall include an AuxPtr field.

9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.

10. Repeat steps 7–9 10 times.

11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.

12. Repeat steps 2–11 for each Round shown in Table 4.14.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3)</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 4)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Advertising_Event_Properties</td>
<td>Data Length</td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0x0001</td>
<td>0x0000</td>
<td>0</td>
<td>191</td>
</tr>
<tr>
<td>3</td>
<td>0x0000</td>
<td>0x0005</td>
<td>191</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 4.14: Payload contents for each case variation.

- **Expected Outcome**

**Pass Verdict**

The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT interlaces multiple advertising events for each set. Each set consists of ADV_EXT_IND PDUs with an optional AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel using the LE Coded PHY.

- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.

- For each set, if advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set using the LE Coded PHY. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs shall not contain any data.

- **Notes**

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.29 LL/DDI/ADV/BV-31-C [Extended Advertising, Multiple Sets, Multiple PHYs, LE 1M and LE Coded PHYs]

- **Test Purpose**

Tests that an advertiser IUT can support multiple advertising sets using both the LE 1M and the LE Coded PHYs with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with and without data are tested.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.
• Reference
[10] 4.4.2.6

• Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map
State: Advertising Parameters Set (selected Adv_INTERVAL_MIN, selected Adv_INTERVAL_MAX, selected
type of advertising events, supported type of own address, selected advertising channel map)

• Test Procedure
Execute the test procedure using multiple types of advertising events with a selected advertising
interval between the minimum and maximum advertising intervals supported using all supported
advertising channels.

For each round from 1 to 7

For each adv. set, Repeat until all data is set

Figure 4.71: LL/DDI/ADV/BV-31-C [Extended Advertising, Multiple Sets, Multiple PHYs, LE 1M and LE Coded
PHYs] – Part A
1. The Upper Tester sends a LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

For each round from 1 to 7 based on Table 4.15:

2. If the Data Length listed in Table 4.15 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 12.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in Table 4.15. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in Table 4.15. The Own_Address_Type shall be
set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).

5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to Table 4.15 and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).

7. The Lower Tester receives interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in Table 4.15. The first set shall use the LE 1M PHY and the second set shall use the LE Coded PHY.

8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the appropriate PHY with the AdvMode field set according to expected properties in Table 4.15 for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in step 5 for the set (if any), it shall include an AuxPtr field.

9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.

10. Repeat steps 7–9 10 times.

11. The Upper Tester both disables advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.

12. Repeat steps 2–11 for each Round shown in Table 4.15.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Advertising_Event_Properties</td>
<td>Data Length</td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0x0001</td>
<td>0x0000</td>
<td>0</td>
<td>191</td>
</tr>
<tr>
<td>3</td>
<td>0x0000</td>
<td>0x0005</td>
<td>191</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0x0000</td>
<td>0x0000</td>
<td>191</td>
<td>191</td>
</tr>
</tbody>
</table>
**Table 4.15:** Payload contents for each case variation.

<table>
<thead>
<tr>
<th>Round</th>
<th>Advertising_Event_Properties</th>
<th>Advertising_Event_Properties</th>
<th>Data Length</th>
<th>Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0x0000</td>
<td>0x0004</td>
<td>251</td>
<td>251</td>
</tr>
<tr>
<td>6</td>
<td>0x0000</td>
<td>0x0001</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0x0000</td>
<td>0x0000</td>
<td>474</td>
<td>0</td>
</tr>
</tbody>
</table>

• **Expected Outcome**

**Pass Verdict**

The IUT returns a Maximum_Advertising_Data_Length between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT interlaces multiple advertising events for each set. Each set consists of ADV_EXT_IND PDUs with an optional AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel. The first set shall use the LE 1M PHY and the second set shall use the LE Coded PHY.

- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.

- For each set, if advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set using the appropriate PHY. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs shall not contain any data.

• **Notes**

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

**4.2.2.30** LL/DDI/ADV/BV-32-C [Extended Advertising, Multiple Sets, Legacy and Extended]

• **Test Purpose**

Tests that an advertiser IUT can support multiple advertising sets using both legacy and extended advertising PDUs in parallel with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with and without data are tested.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.
• Reference

[10] 4.4.2.6

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

• Test Procedure

Execute the test procedure using multiple types of advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.

For each round from 1 to 7

- HCI_LE_Set_Extended_Advertising_Parameters (Event Properties, First Set, LE 1M PHY)
- HCI_Command_Complete_Event (Status: 0x00)

For each adv. set, Repeat until all data is set

- HCI_LE_Set_Extended_Advertising_Data
- HCI_Command_Complete_Event (Status: 0x00)

- HCI_LE_Set_Extended_Advertising_Enable (First Set, Enable)
- HCI_Command_Complete_Event (Status: 0x00)

- HCI_LE_Set_Extended_Advertising_Enable (Second Set, Enable)
- HCI_Command_Complete_Event (Status: 0x00)

**Figure 4.73: LL/DDI/ADV/BV-32-C [Extended Advertising, Multiple Sets, Legacy and Extended] – Part A**
1. The Upper Tester sends a LE_Read_Maximum_Advertising_Data_Length command to the IUT and expects the IUT to return a Maximum_Advertising_Data_Length between 0x001F and 0x0672. The Upper Tester stores the Maximum_Advertising_Data_Length for future use.

For each round from 1 to 7 based on Table 4.16:

2. If the Data Length listed in Table 4.16 for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 12.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in Table 4.16. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the
minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in Table 4.16. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

5. The Upper Tester sends zero or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to Table 4.16 and using random octets from 1 to 255 as the payload. If the Data Length is zero for a given set, the Upper Tester shall do nothing for that set. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).

6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, if the Advertising_Event_Properties parameter was set to 0x1D in step 3 or 4, then Duration[i] is set to 0x0080 (1280ms); otherwise the Duration[i] parameter is set to 0x0000 (No Advertising Duration).

7. The Lower Tester receives interlaced advertising events for each set, consisting of either legacy PDUs or ADV_EXT_IND PDUs from the IUT with the AdvMode set according to expected properties in Table 4.16 for each set.

8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in Table 4.16 for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in step 5 for the set (if any), it shall include an AuxPtr field.

9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.

10. Repeat steps 7–9 10 times.

11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.

12. The Upper Tester clears the advertising sets using one HCI_LE_Clear_Advertising_Sets command.

13. Repeat steps 2–11 for each Round shown in Table 4.16.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 3)</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 4)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5)</th>
<th>HCI_LE_Set_Extended_Advertising_Data (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>Advertising_Event_Properties</td>
<td>Data Length</td>
<td>Data Length</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x0010</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0x0013</td>
<td>0x0000</td>
<td>0</td>
<td>191</td>
</tr>
<tr>
<td>Round</td>
<td>HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set</td>
<td>HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set</td>
<td>HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set</td>
<td>HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>Advertising_Event_Properties 0x0000</td>
<td>Advertising_Event_Properties 0x0013</td>
<td>Data Length 191</td>
<td>Data Length 31</td>
</tr>
<tr>
<td>4</td>
<td>Advertising_Event_Properties 0x0001</td>
<td>Advertising_Event_Properties 0x0010</td>
<td>Data Length 31</td>
<td>Data Length 31</td>
</tr>
<tr>
<td>5</td>
<td>Advertising_Event_Properties 0x0003</td>
<td>Advertising_Event_Properties 0x0004</td>
<td>Data Length 0</td>
<td>Data Length 251</td>
</tr>
<tr>
<td>6</td>
<td>Advertising_Event_Properties 0x0000</td>
<td>Advertising_Event_Properties 0x001D</td>
<td>Data Length 253</td>
<td>Data Length 0</td>
</tr>
<tr>
<td>7</td>
<td>Advertising_Event_Properties 0x0000</td>
<td>Advertising_Event_Properties 0x0012</td>
<td>Data Length 474</td>
<td>Data Length 0</td>
</tr>
</tbody>
</table>

Table 4.16: Payload contents for each case variation.

• Expected Outcome

**Pass Verdict**

The IUT returns a **Maximum_Advertising_Data_Length** between 0x001F and 0x0672.

For all rounds described in the test procedure, the following condition shall occur:

- The IUT interlaces multiple advertising events for each set. If the set utilizes legacy PDUs, the IUT sends the appropriate legacy PDU, otherwise each set consists of ADV_EXT_IND PDUs with an optional AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.

- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.

- For each set, if advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set, or using legacy PDUs if expected for the given set. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs or legacy PDUs shall not contain any data.

• Notes

The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.31   LL/DDI/ADV/BV-33-C [Extended Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)]

• Test Purpose

Tests that an advertiser IUT can support multiple periodic advertising sets on the same PHY or different PHYs.
The Lower Tester observes the event timing and packet contents on the advertising channels in use. The Lower Tester confirms that Channel Selection Algorithm #2 is utilized for the periodic advertisements.

- **Reference**
  
  [10] 4.4.2.2.2, 2.3.4.5

- **Initial Condition**
  
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

  The number of simultaneous advertising sets supported is specified in the “Supported Simultaneous Advertising Sets” IXIT parameter.

- **Test Procedure**

  - For each applicable round from 1 to 12
  
    Send each command for all Advertising Sets specified in the round

    - **HCI LE Set Extended Advertising Parameters**
      
      (Extended Advertising)
      
      **HCI Command Complete Event**
      
      (Status: 0x00)

    - **HCI LE Set Periodic Advertising Parameters**
      
      **HCI Command Complete Event**
      
      (Status: 0x00)

    - **HCI LE Set Periodic Advertising Enable**
      
      (Enable)
      
      **HCI Command Complete Event**
      
      (Status: 0x00)

    - **HCI LE Set Extended Advertising Enable**
      
      (Enable)
      
      **HCI Command Complete Event**
      
      (Status: 0x00)

  - **ADV EXT IND**
    
    (AdvMode: 00b, AuxPTr)
    
    (AdvMode: 00b, SyncInf)
    
    AUX ADV IND
  
    AUX_SYNC_IND

  - **For All Advertising Sets Used in This Round**

  - **REPEAT 100 TIMES**

  - **HCI LE Set Extended Advertising Enable**
    
    (Disable, First Set)
    
    **HCI Command Complete Event**
    
    (Status: 0x00)

  **Figure 4.75:** LL/DDI/ADV/BV-33-C [Extended Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)] – Part A
For each applicable round in Table 4.17 below:

1. The Upper Tester sends HCI_LE_Set_Extended_Advertising_Parameters commands to the IUT to create the specified number of advertising sets using all supported advertising channels and 50ms advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The PHY for primary and secondary advertising channels shall be as specified in the table. The Upper Tester receives an HCI_Command_Complete event from the IUT for each command with Status set to 0x00 (Success).

2. For each set, the Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels. Periodic_Advertising_Properties parameter shall be set to 0x0000. For each round, the advertising intervals for the first, second and third sets shall be set to, respectively, 15ms, 20ms and 22.5ms. The Upper Tester receives an HCI_Command_Complete event from the IUT for each command with Status set to 0x00 (Success).

3. For each set, the Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (Periodic Advertising).

4. The Upper Tester enables each advertising set using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).

5. The Lower Tester scans for each advertising PHY supported by the IUT. For each set, the Lower Tester receives interlaced advertising events from the IUT using the Primary Advertising PHY for...
which that set was configured in step 1, consisting of ADV_EXT_IND packets with AdvMode set to 00b with the AuxPtr Extended Header field present.

6. For each set, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the Secondary Advertising PHY for which that set was configured in step 1, with the AdvMode field set to 00b and the SyncInfo Extended Header fields present.

7. Repeat steps 5–6 100 times. At the same time, for each set, the Lower Tester utilizes the SyncInfo field to listen for AUX_SYNC_IND PDUs on the secondary advertising channel using the Secondary Advertising PHY for which that set was configured in step 1 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDUs shall have the AdvMode field set to 00b.

8. The Upper Tester disables extended advertising for the first set using the HCI_LE_Set_Extended_Advertising_Enable command but maintains synchronization with the IUT’s periodic advertising.

9. The Lower Tester confirms that periodic advertising continues for the first set and that both extended advertising and periodic advertising continue for the remaining set(s) by repeating steps 5–7, except that in steps 5–6, the Lower Tester only receives advertising PDUs for the remaining set(s).

10. The Upper Tester disables extended advertising for the remaining set(s) using the HCI_LE_Set_Extended_Advertising_Enable command but maintains synchronization with the IUT’s periodic advertising.

11. For both sets on each supported advertising PHY, the Lower Tester maintains synchronization to periodic advertising and receives at least 100 AUX_SYNC_IND PDUs on the secondary advertising using the indices selected by the LE Channel Selection Algorithm #2 using the Secondary Advertising PHY for which that set was configured in step 1.

12. The Upper Tester disables periodic advertising on all sets using the HCI_LE_Set_Periodic_Advertising_Enable command.

Table 4.17 specifies the rounds to be carried out based on the supported PHYs and the maximum number of supported advertising sets. Round 1 is always carried out; the remaining rounds are only carried out if a PHY other than LE 1M is supported.

<table>
<thead>
<tr>
<th>Round</th>
<th>Supported PHYSs and max sets</th>
<th>First set</th>
<th>Second set</th>
<th>Third set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2M ≥ 2</td>
<td>Coded ≥ 2</td>
<td>Both 2</td>
<td>Both ≥ 3</td>
</tr>
<tr>
<td>1</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>2</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>3</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>4</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>5</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>6</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>7</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Round</td>
<td>Supported PHYs and max sets</td>
<td>First set</td>
<td>Second set</td>
<td>Third set</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------</td>
<td>-----------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>2M ≥ 2 Coded ≥ 2 Both ≥ 2 Both ≥ 3</td>
<td>Primary PHY</td>
<td>Secon’y PHY</td>
<td>Primary PHY</td>
</tr>
<tr>
<td>8</td>
<td>•</td>
<td>1M</td>
<td>2M</td>
<td>Coded</td>
</tr>
<tr>
<td>9</td>
<td>•</td>
<td>Coded</td>
<td>Coded</td>
<td>1M</td>
</tr>
<tr>
<td>10</td>
<td>•</td>
<td>1M</td>
<td>1M</td>
<td>1M</td>
</tr>
<tr>
<td>11</td>
<td>•</td>
<td>1M</td>
<td>2M</td>
<td>Coded</td>
</tr>
<tr>
<td>12</td>
<td>•</td>
<td>Coded</td>
<td>Coded</td>
<td>1M</td>
</tr>
</tbody>
</table>

Table 4.17: Specification of rounds for LL/DDI/ADV/BV-33-C

• Expected Outcome

  **Pass Verdict**
  
  For each advertising set specified in the test procedure, the following condition shall occur:
  
  - The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel using the advertising PHY for which that set was configured.
  
  - The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.
  
  - The IUT sends AUX_SYNC_IND PDUs with correct timings on the secondary advertising channel using indices selected by the Channel Selection Algorithm #2.

  **Inconclusive Verdict**
  
  - The IUT fails to set extended advertising parameters or periodic advertising parameters for any of the specified advertising sets and returns a Memory Capacity Exceeded Error.

• Notes

  The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.32 LL/DDI/ADV/BV-33-C [Extended Advertising, TX Power with RF Path Compensation]

• Test Purpose

  Tests that an advertiser IUT can report the TX Power in advertisements with RF path compensation using correct payload fields, timing, and channel sequence for the maximum time allowed.

• Reference

  [10] 2.3.4.7, 4.4.2.6

• Initial Condition

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Non-Connectable Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)
For each round from 1 to 4 based on Table 4.18:

1. The Upper Tester sends an HCI_LE_Write_RF_Path_Comensation command to the IUT. The RF_Tx_Path_Comensation_Value shall be set as shown in Table 4.18 for the current Round.
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter bit 6 (Include TxPower in the advertising PDU) shall be set and all other bits cleared. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M).
3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
4. The Lower Tester expects the IUT to send ADV_EXT_IND on the first supported advertising channel. The AdvMode shall be set to 00b. If an AuxPtr field is present, the Lower Tester expects the IUT to send an AUX_ADV_IND on the specified secondary channel with the AdvMode set to 00b and that has a TxPower field present; otherwise the ADV_EXT_IND shall have a TxPower field present. The TxPower field shall contain a value adjusted for the RF_Tx_Path_Comensation_Value used in this round, relative to the unadjusted value received in Round 1.
5. Repeat step 4 until 100 advertising events have been detected.
6. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.
7. Repeat steps 1–6 for each Round shown in Table 4.18.
<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Write_RF_Path_Compensation (Step 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RF Tx Path Compensation Value</td>
</tr>
<tr>
<td>1</td>
<td>0 dB</td>
</tr>
<tr>
<td>2</td>
<td>+5 dB</td>
</tr>
<tr>
<td>3</td>
<td>-5 dB</td>
</tr>
<tr>
<td>4</td>
<td>0 dB</td>
</tr>
</tbody>
</table>

_Table 4.18: RF Tx Path Compensation Values._

- Expected Outcome
  - **Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:
- The Lower Tester receives an ADV_EXT_IND with a Tx Power field.
- The Tx Power field contains a value adjusted for the RF_Tx_Path_Compensation_Value used relative to the unadjusted value received in Round 1.

4.2.2.33 **LL/DDI/ADV/BV-35-C [Extended Advertising, Multiple Sets, Maximum Supported Sets]**

- **Test Purpose**
  Tests that an advertiser IUT can support multiple advertising sets with the correct payload fields, timing, and channel sequence for the maximum time allowed. Advertisements with the minimum data required to be supported are tested.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used.

- **Reference**
  [10] 4.4.2.6

- **Initial Condition**
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)

- **Test Procedure**
  Execute the test procedure with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.
1. The Upper Tester sends TSPX_adv_sets_max HCI_LE_Set_Extended_Advertising_Parameters commands to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M). The Advertising_SID shall be set to a random value, [0x00...0x0F], for each set.

2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command to the IUT for each advertising set, with length 31 and using random octets from 1 to 255 as the payload.

3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command for each set. The Duration parameter is set to 0x0000 (No Advertising Duration).

4. The Lower Tester receives interleaved advertising events for each set, consisting of ADV_EXT_IND PDUs from the IUT. The ADV_EXT_IND PDUs shall contain an AuxPtr field.

5. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY. If the AUX_ADV_IND PDU does not contain all the data submitted in step 2 for the set, it shall include an AuxPtr field.

6. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU containing additional data submitted in step 2 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.

7. Repeat steps 4–6 10 times.

8. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command for each set.
• Expected Outcome

Pass Verdict
The IUT interlaces multiple advertising events for each set. Each set consists of ADV_EXT_IND PDUs with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
For each set, the Lower Tester receives all the data sent in step 2, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set.

• Notes
The notes in Section 4.2.2.2 describe the reasoning of the timing measurements and test result criteria.

4.2.2.34 AoD Connectionless CTE Advertising

• Test Purpose
Tests that an advertiser IUT can send advertising packets with the AoD Connectionless Constant Tone Extension included with correct contents and with the correct event timing when utilizing a public device address. Advertisements without data, along with chaining, are tested.
The Lower Tester is configured to scan for an AoD Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension data. The Lower Tester observes the event timing and packet contents on the advertising channels in use.

• Reference
[13] 2.5.2, 2.5.3

• Initial Condition
Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map
State: Advertising Parameters Set (selected Adv_INTERVAL, selected Adv_INTERVAL, selected type of advertising events, supported type of own address, selected advertising channel map)
The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.
The maximum number of packets with CTE to be transmitted in a periodic advertising event is defined by the TSPX_max_CTE_count IXIT entry and the corresponding periodic advertising interval is defined by the TSPX_per_adv_interval IXIT entry.
### Test Procedure

For each round from 1 to 3

- **Lower Tester**
  - `HCI_LE_Set_Periodic_Advertising_Parameters`
  - `HCI_Command_Complete_Event`
    - (Status: 0x00)
  - `HCI_LE_Set_Periodic_Advertising_Enable`
  - `HCI_Command_Complete_Event`
    - (Status: 0x00)

- **IUT**
  - `HCI_LE_Set_Extended_Advertising_Parameters`
    - (Extended Advertising)
  - `HCI_Command_Complete_Event`
    - (Status: 0x00)
  - `HCI_LE_Set_Periodic_Advertising_Parameters`
  - `HCI_Command_Complete_Event`
    - (Status: 0x00)
  - `HCI_LE_Set_Periodic_Advertising_Enable`
  - `HCI_Command_Complete_Event`
    - (Enable)
  - `HCI_LE_Set_Periodic_Advertising_Enable`
    - (Enable)

- **Upper Tester**

Repeat for 1 µs and 2 µs slots (where supported)

---

**Figure 4.79: AoD Connectionless CTE Advertising – Part A**
1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_Length_Switching_Pattern between 0x02 and 0x4B and a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_Length_Switching_Pattern and the Max_CTE_Length for future use.

For each round from 1 to 3 based on **Table 4.19:**

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>Max_CTE_Length from step 1</td>
</tr>
</tbody>
</table>

**Table 4.19: Parameter values for each case variation**

2. If the CTETime listed in **Table 4.19** for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 17.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall
be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M and the Secondary_Advertising_PHY shall be set to the value as specified in Table 4.20.

4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and periodic advertising interval set to TSPX_per_adv_interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.

5. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 4. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDS[0] through Antenna_IDS[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, …, TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDS[] values. CTE_Type shall be set as specified in Table 4.20. CTE_Length shall be set to the value specified in Table 4.19. CTE_Count shall be set to TSPX_max_CTE_count.

6. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).

7. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).

8. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).

9. The Lower Tester receives an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.

10. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.

11. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType set as specified in Table 4.20. The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5.

12. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType as specified in Table 4.20. The packet containing the AUX_CHAIN_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field.

13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

14. The Lower Tester confirms that periodic advertising continues with CTE Transmission when extended advertising is disabled by repeating steps 11 and 12.

15. The Upper Tester disables CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command.

16. The Lower Tester confirms that periodic advertising continues without CTE Transmission when CTE Transmission is disabled by repeating steps 11 and 12, except that the packet containing the AUX_SYNC_IND PDU shall not contain the Constant Tone Extension field and the AUX_SYNC_IND PDU shall not contain the CTEInfo field or the AuxPtr field.
17. The Upper Tester disables periodic advertising using the 
HCI_LE_Set_Periodic_Advertising_Enable command.
18. Repeat steps 2–17 for each round shown in Table 4.19.

- Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Secondary Advertising PHY</th>
<th>CTE Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2.34.1 LL/DDI/ADV/BV-36-C [AoD Connectionless CTE Advertising – LE 1M PHY, 2 µs slots]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (2 µs slots)</td>
</tr>
<tr>
<td>4.2.2.34.2 LL/DDI/ADV/BV-57-C [AoD Connectionless CTE Advertising – LE 2M PHY, 2 µs slots]</td>
<td>0x02 (LE 2M PHY)</td>
<td>0x02 (2 µs slots)</td>
</tr>
<tr>
<td>4.2.2.34.3 LL/DDI/ADV/BV-58-C [AoD Connectionless CTE Advertising – LE 1M PHY, 1 µs slots]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (1 µs slots)</td>
</tr>
<tr>
<td>4.2.2.34.4 LL/DDI/ADV/BV-59-C [AoD Connectionless CTE Advertising – LE 2M PHY, 1 µs slots]</td>
<td>0x02 (LE 2M PHY)</td>
<td>0x01 (1 µs slots)</td>
</tr>
</tbody>
</table>

Table 4.20: AoD Connectionless CTE Advertising Test Cases

- Expected Outcome

  Pass Verdict

  For all rounds described in the test procedure, the following condition shall occur:

  - The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
  - The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.
  - When CTE Transmission is enabled, the packets containing AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs and AUX_CHAIN_IND PDUs include the CTEInfo field.
  - When CTE Transmission is disabled, the packets containing AUX_SYNC_IND PDUs do not include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs do not include the CTEInfo field or the AuxPtr field.
  - The IUT sends a number of CTE_Count packets containing a Constant Tone Extension in each periodic advertising event.
4.2.2.35 AoA Connectionless CTE Advertising

- Test Purpose
  Tests that an advertiser IUT can send advertising packets with the AoA Connectionless Constant Tone Extension included with correct contents and with the correct event timing when utilizing a public device address. Advertisements without data, along with chaining, are tested.

  The Lower Tester is configured to scan for an AoA Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension. The Lower Tester observes the event timing and packet contents on the advertising channels in use.

- Reference
  [13] 2.5.2, 2.5.3

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Advertising Parameters Set (selected Adv_Interval, selected Adv_Interval, selected type of advertising events, supported type of own address, selected advertising channel map)

  The maximum number of packets with CTE to be transmitted in a periodic advertising event is defined by the TSPX_max_CTE_count IXIT entry and the corresponding periodic advertising interval is defined by the TSPX_per_adv_interval IXIT entry.
• Test Procedure

![Diagram showing test procedure for Link Layer (LL) test suite with steps for HCI_LE_Set_Extended_Advertising_Parameters, HCI_Command_Complete_Event, HCI_LE_Set_Connectionless_CTE_Transmit_Parameters, and HCI_Command_Complete_Event.]

For each round from 1 to 3

- HCI_LE_Set_Periodic_Advertising_Parameters (Status: 0x00)
- HCI_Command_Complete_Event

Figure 4.81: AoA Connectionless CTE Advertising – Part A
1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_CTE_Length for future use.

For each round from 1 to 3 based on Table 4.21:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>Max_CTE_Length from step 1</td>
</tr>
</tbody>
</table>

Table 4.21: Parameter values for each case variation

2. If the CTETime listed in Table 4.21 for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 17.

3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall
be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M) and the Secondary_Advertising_PHY shall be set to the value specified in Table 4.22.

4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and periodic advertising interval set to TSPX_per_adv_interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.

5. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 4. CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). CTE_Length shall be set to the value specified in Table 4.21. CTE_Count shall be set to TSPX_max_CTE_count.

6. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).

7. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).

8. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).

9. The Lower Tester receives an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.

10. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.

11. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisments. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to ‘0’, and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5.

12. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to ‘0’, and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing the AUX_CHAIN_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field.

13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

14. The Lower Tester confirms that periodic advertising continues with CTE Transmission when extended advertising is disabled by repeating steps 11 and 12.

15. The Upper Tester disables CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command.

16. The Lower Tester confirms that periodic advertising continues without CTE Transmission when CTE Transmission is disabled by repeating steps 11 and 12, except that the packet containing the AUX_SYNC_IND PDU shall not contain the Constant Tone Extension field and the AUX_SYNC_IND PDU shall not contain the CTEInfo field or the AuxPtr field.

17. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

18. Repeat steps 2–17 for each round shown in Table 4.21.
### Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Secondary Advertising PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2.35.1 LL/DDI/ADV/ BV-37-C [AoA Connectionless CTE Advertising – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.2.2.35.2 LL/DDI/ADV/ BV-60-C [AoA Connectionless CTE Advertising – LE 2M PHY]</td>
<td>0x02 (LE 2M PHY)</td>
</tr>
</tbody>
</table>

*Table 4.22: AoD Connectionless CTE Advertising Test Cases*

### Expected Outcome

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.

- The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.

- When CTE Transmission is enabled, the packets containing AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs and AUX_CHAIN_IND PDUs include the CTEInfo field.

- When CTE Transmission is disabled, the packets containing AUX_SYNC_IND PDUs do not include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs do not include the CTEInfo field or the AuxPtr field.

- The IUT sends a number of CTE_Count packets containing a Constant Tone Extension in each periodic advertising event.

### 4.2.2.36 LL/DDI/ADV/BV-39-C [Connectionless CTE Advertising – Maintain CTE Configuration]

**Test Purpose**

Tests that an advertiser IUT configured to send advertising packets with Constant Tone Extension and with disabled periodic advertisements starts sending Constant Tone Extensions after periodic advertisements are re-enabled.

The Lower Tester is configured to scan for Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension.

**Reference**

[13] 2.5.2, 2.5.3

**Initial Condition**

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Advertising Parameters Set (selected Adv.Interval, selected Adv.Interval, selected type of advertising events, supported type of own address, selected advertising channel map)
• Test Procedure

Figure 4.83: LL/DDI/ADV/BV-39-C [Connectionless CTE Advertising - Maintain CTE Configuration] – Part A
1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14.

2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.

4. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 3. CTE_Type and CTE_Length shall be set to a valid value supported by the IUT.

5. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).
6. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).

7. The Upper Tester enables advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).

8. The Lower Tester receives an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.

9. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.

10. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 4, RFU set to '0', and the CTE_Type set to the CTE_Type value from step 4. The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 4.

11. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

12. The Lower Tester confirms that extended advertising continues by repeating step 8.

13. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall not contain the SyncInfo or the CTEInfo field.

14. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).

15. The Lower Tester confirms that extended advertising continues by repeating steps 8–10.

16. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

17. The Upper Tester disables extended advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command.

**Expected Outcome**

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.

- When periodic advertisement is enabled, the AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND.

- When periodic advertisement is disabled, the AUX_ADV_IND does not include a SyncInfo field.

- The packets containing AUX_SYNC_IND PDUs include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs include the CTEInfo field.

4.2.2.37 LL/DDI/ADV/BV-43-C [Periodic Advertising validating SyncInfo fields]

**Test Purpose**

Tests that the AA and CRCInit fields shall be the same for all SyncInfo fields describing the same periodic advertising. Non-Connectable and Non-Scannable advertisements are tested. The Lower
Tester confirms that the AA and CRCInit fields in all SyncInfo fields for the same periodic advertising train are identical.

- **Reference**
  
  [10] 4.2.2.24

- **Initial Condition**
  
  State: Advertising Parameters Set (supported type of own address, selected advertising channel map)

- **Test Procedure**
  
  Execute the test procedure using non-connectable and non-scannable periodic advertising events with a selected periodic advertising interval. Confirm that the AUX_ADV_IND packets received by the Lower Tester have identical AA and CRCInit fields in the SyncInfo parameter.

![Diagram](image)

**Figure 4.85: LL/DDI/ADV/BV-43-C [Periodic Advertising validating SyncInfo fields]**

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The
Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

1. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using minimum advertising interval. Periodic_Advertising_Properties parameter is set to 0x0000.

2. The Upper Tester sends one HCI_LE_Set_Periodic_Advertising_Data command to the IUT with a data length of 0x01 and value of 0x00.

3. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (Periodic Advertising).

4. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

5. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present.

6. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields present. Note: There can be AUX_SYNC_IND PDUs transmitted forming the periodic advertising.

7. Repeat steps 6–7 10 times.

• Expected Outcome

  Pass Verdict

  - The Lower Tester confirms the AA and CRCInit fields in the SyncInfo extended payload are identical for each SyncInfo field in each AUX_ADV_IND PDU.

4.2.2.38 LL/DDI/ADV/BI-05-C [Disallow Extended Advertising PDU sizes for Legacy Advertising when advertising enabled]

• Test Purpose

  Verify that a supported legacy command yields a Command Complete event with status ‘Invalid HCI Command Parameters’ in return when sending PDU sizes larger than legacy advertising PDU after advertising has been enabled.

• Reference

  [1] 7.8.54, 7.8.56

• Initial Condition

  State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected type of advertising events, supported type of own address, selected advertising channel map)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command with Advertising_Event_Properties set to "Use legacy advertising PDUs".
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command with Data Length equal to 31 to the IUT. The IUT sends a Command Complete event with Status set to 0x00.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable to the IUT. The IUT is expected to return an HCI_Command_Complete event with Error Code 0x00 (Success).
4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command with Data Length equal to 32 to the IUT. The IUT sends a Command Complete event with Status set to 0x12.

• Expected Outcome

Pass Verdict

The IUT returns an HCI Command Complete event with Status = Invalid HCI Command Parameters after the HCI_LE_Set_Extended_Advertising_Data command in step 4 is received.

4.2.2.39 LL/DDI/ADV/BI-06-C [Disallow Extended Advertising PDU sizes for Scannable Legacy Advertising when advertising enabled]

• Test Purpose

Verify that the HCI_LE_Set_Extended_Scan_Response_Data command yields a Command Complete event with status 'Invalid HCI Command Parameters' in return when sending scannable legacy PDU sizes larger than legacy advertising PDU after advertising has been enabled.

• Reference

[1] 7.8.55, 7.8.56
• **Initial Condition**

State: Advertising Parameters Set (selected Adv.Interval.Min, selected Adv.Interval.Max, selected type of advertising events, supported type of own address, selected advertising channel map)

• **Test Procedure**

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command with Advertising_Event_Properties set to “Scannable Legacy advertising” and “Use legacy advertising PDUs”.
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Response_Data command with Data Length equal to 31 to the IUT. The IUT sends a Command Complete event with Status set to 0x00.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable to the IUT. The IUT is expected to return an HCI_Command_Complete event with Error Code 0x00 (Success).
4. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Response_Data command with Data Length equal to 32 to the IUT. The IUT sends a Command Complete event with Status set to 0x12.

• **Expected Outcome**

Pass Verdict

The IUT returns an HCI Command Complete event with Status = Invalid HCI Command Parameters after the HCI_LE_Set_Scan_Response_Data command in step 4 is received.

4.2.2.40 LL/DDI/ADV/BV-61-C [Extended Advertising, Periodic Advertising with TxPower]

• **Test Purpose**

Tests that an advertiser IUT correctly reports the TxPower in periodic advertisements.

• **Reference**

[10] Section 2.3.4.7, 4.4.2.6, 4.4.2.12
• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Standby

• Test Procedure

```
Lower Tester   IUT        Upper Tester

HCI_LE_Set_Extended_Advertising_Parameters
(Extended Advertising)

HCI_Command_Complete_Event
(Status: 0x00)

HCI_LE_Set_Periodic_Advertising_Parameters

HCI_Command_Complete_Event
(Status: 0x00)

HCI_LE_Set_Periodic_Advertising_Data

HCI_Command_Complete_Event
(Status: 0x00)

Repeat until all data is set

HCI_LE_Set_Periodic_Advertising_Enable
(Enable)

HCI_Command_Complete_Event
(Status: 0x00)

HCI_LE_Set_Extended_Advertising_Enable
(Enable)

HCI_Command_Complete_Event
(Status: 0x00)
```

Figure 4.88: LL/DDI/ADV/BV-61-C [Extended Advertising, Periodic Advertising with TxPower] – Part A
Figure 4.89: LL/DDI/ADV/BV-61-C [Extended Advertising, Periodic Advertising with TxPower] – Part B

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. The Advertising_Event_Properties parameter is set to 0. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M).

2. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. The Periodic_Advertising_Properties parameter bit 6 (Include TxPower in the advertising PDU) shall be set and all other bits cleared.

3. The Upper Tester sends enough HCI_LE_Set_Periodic_Advertising_Data commands to the IUT to ensure that the IUT will need to send at least one AUX_CHAIN_IND PDU.
4. The Upper Tester enables periodic advertising using the 
   HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01  
   (Periodic Advertising).
5. The Upper Tester enables advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable  
   command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
6. The Lower Tester expects the IUT to send ADV_EXT_IND on the first supported advertising  
   channel. The AdvMode shall be set to 00b.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary  
   advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields  
   present.
8. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the  
   secondary advertising channel using the index selected by the LE Channel Selection Algorithm  
   #2 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall have the  
   AdvMode field set to 00b, shall contain an AuxPtr field and shall include a TxPower field.
9. The Lower Tester utilizes the previous AuxPtr field to listen for an AUX_CHAIN_IND PDU with  
   the AdvMode field set to 00b and containing additional data submitted in Step 5. The Lower  
   Tester expects the AUX_CHAIN_IND PDU not to contain a TxPower field. This step is repeated  
   until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.
10. Repeat Steps 5–8 10 times.
11. The Upper Tester disables extended advertising using the 
    HCI_LE_Set_EXTENDED_Advertising_Enable command but maintains synchronization with the  
    IUT's periodic advertising.
12. The Lower Tester confirms that periodic advertising with the expected TxPower field continues  
    when extended advertising is disabled by repeating steps 7–8 10 times.
13. The Upper Tester disables periodic advertising using the 
    HCI_LE_Set_Periodic_Advertising_Enable command.

• Expected Outcome

Pass Verdict
The Lower Tester receives all AUX_SYNC_IND PDUs with a TxPower field.
The Lower Tester receives all AUX_CHAIN_IND PDUs without a TxPower field.

4.2.2.41 LL/DDI/ADV/BV-62-C [Periodic Advertising, Channel Map Update]

• Test Purpose
Tests that an advertiser IUT sending periodic advertisements correctly provides the channel map  
update indication.

• Reference
[10] 4.4.2.13.1
[18] 1.20

• Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,  
LL_advertiser_Adv_Channel_Map
State: Advertising Parameters Set (selected Adv_Interval_Min, selected Adv_Interval_Max, selected  
type of advertising events, supported type of own address, selected advertising channel map)  
All primary and secondary channels are enabled in the channel map.
• **Test Procedure**

Execute the test procedure using periodic advertising events with a selected periodic advertising interval between the minimum and maximum advertising intervals supported.

![Diagram of test procedure](image)

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set to the LE 1M PHY.
2. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and selected periodic interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.
3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Data command to the IUT with Advertising_Data_Length set to 31. The payload shall consist of random octets from 1 to 255.
4. The Upper Tester enables periodic advertising using the 
   HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 
   (Periodic Advertising).
5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable 
   command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
6. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with 
   the AuxPtr Extended Header field present.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary 
   advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields 
   present.
8. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the 
   secondary advertising channel using the index selected by the LE Channel Selection Algorithm 
   #2 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall have the 
   AdvMode field set to 00b.
9. The Lower Tester receives data as specified in step 3.
10. The Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT 
    setting the channel map to only use even channels. The Upper Tester receives an 
    HCI_Command_Complete event from the IUT in response.
11. The Lower Tester receives AUX_SYNC_IND PDUs with the Channel Map Update Indication data 
    present only in the extended header prior to the instant.
12. The Lower Tester continues to receive periodic data using channelMap\textsubscript{NEW} after the instant.

- Expected Outcome

  **Pass Verdict**
  The IUT provides channelMap\textsubscript{NEW} in the extended header a minimum of 6 AUX_SYNC_IND PDUs 
  before the instant occurs.
  The Lower Tester receives data from the IUT utilizing the new channel map from the instant.

  **Inconclusive Verdict**
  The IUT accepts the HCI_LE_Set_Host_Channel_Classification command but chooses not to update 
  the channel map and no Channel Map Update indication is provided.

4.2.3 **SCN**
Tests that the IUT behaves according to the device discovery procedures in the scanner role.

4.2.3.1 **Common PDU Contents**
The default advertising channel packet contents for the Lower Tester sent and Upper Tester accepted 
packets are defined in `/LL/DDI`.

The addresses used vary per test and the data is typically a count of the events transmitted by the Lower 
Tester. The Lower Tester uses the device name “LT” in scan response packets.

4.2.3.2 **LL/DDI/SCN/BV-01-C [Passive Scanning: Non-connectable]**

- **Test Purpose**
  Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using non-connectable advertising events on one channel at a time and 
expects the IUT to report the advertising by the Lower Tester. The advertising packets’ payload is a 
sequence numbering of the packets.
• Reference

[3] 4.4.3.1

• Initial Condition

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

• Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

1. Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
2. Upper Tester enables passive scanning in the IUT.
3. Lower Tester sends ADV_NONCONN_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
4. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in step 3 and data included in one of the advertising packets.
5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

Figure 4.91: LL/DDI/SCN/BV-01-C [Passive Scanning: Non-connectable]
6. Configure Lower Tester as advertiser using the second supported advertising channel and a common public address that differs from the IUT address in the least significant octet.
7. Repeat steps 2–5.
8. Configure Lower Tester as advertiser using the third supported advertising channel using a common public address that differs from the IUT address in the most significant octet and a random address type.
9. Repeat steps 2–5.

- Expected Outcome

Pass Verdict
The test procedure completes using each supported advertising channel separately, with the IUT recognizing each advertising packet type.

4.2.3.3 LL/DDI/SCN/BV-02-C [Passive Scanning Device Filtering]

- Test Purpose
Tests that a scanner IUT is able to scan for a specific advertiser using device filtering.

The Lower Tester advertises using non-connectable events on one channel and expects the IUT to report the information in the advertising packets. The advertising packets' payload is a sequence numbering of the packets sent.

- Reference
[3] 4.3.3, 4.4.3.1

- Initial Condition
Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX.
State: Passive Scanning (selected scan interval, selected scan window) AND Specific White Listed (one white listed device address, public address type, policy for scanner, black list all unknown devices)

- Test Procedure
Execute the test procedure using the minimum advertising interval and the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval. Use the filtering policy to black list all unknown devices (Ignore advertising packets from devices not in the White List Only (0x01) or with an incorrect address type).
1. Upper Tester enables passive scanning in the IUT with the filtering policy set to ‘Ignore advertising packets from devices not in the White List Only (0x01)’.
2. Configure Lower Tester as advertiser, using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
3. Lower Tester sends ADV_NONCONN_IND packets each advertising event on the selected advertising channel only. Repeat until 20 scan intervals have passed or step 4 executes.
4. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT which indicates use of non-connectable advertising events, a public address type, and the white listed device address.
5. Disable advertising in Lower Tester.
6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a random address type.
7. Lower Tester sends ADV_NONCONN_IND packets each advertising event on the selected advertising channel only. Repeat until 20 scan intervals have passed.
8. Disable advertising in Lower Tester.
9. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type. The device address shall differ from the IUT address in the least significant octet.
10. Repeat steps 7–8.
11. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a random address type. The device address shall differ from the IUT address in the most and least significant octets.
12. Repeat steps 7–8.
13. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and receive an HCI_Command_Complete event in response.

• Expected Outcome
  Pass Verdict
  The test procedure completes using a specific advertising channel with the timing combination.
  The IUT reports the device address matching the white listed both for the address and type with an HCI event,
  The IUT does not report the device address matching the white listed for the address and but mismatching the type with an HCI event,
  The IUT does not report the device addresses mismatching the white listed address with an HCI event.

4.2.3.4 LL/DDI/SCN/BV-03-C [Active Scanning]

• Test Purpose
  Tests that a scanner IUT detects and requests additional information from advertisers and reports the results from the Controller.
  The Lower Tester advertises using connectable undirected events on one channel at a time, acting as multiple advertisers, expects the IUT to send the request, then responds to the IUT. The advertising packets' payload is a sequence numbering of the packets sent. The Lower Tester observes the packets and timing of the IUT, as well as the HCI events reporting scan results.

• Reference
  [3] 4.4.3.2

• Initial Condition
  State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))

• Test Procedure
  Execute the test procedure advertising on a selected advertising channel at a time, using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.
1. Configure Lower Tester advertising channel map using the first supported advertising channel.
2. Configure Lower Tester as advertiser using a common device address and a public address type.
3. Upper Tester enables active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.
4. Lower Tester sends ADV_IND packets with event count as data encoded unsigned least significant bit first, each advertising event on the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20) or step 5 executes.
5. Lower Tester receives an SCAN_REQ packet T_IFS after any of the ADV_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after of the SCAN_REQ packet.
6. Interleave with step 4: Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the ADV_IND packets.
8. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.
9. Configure Lower Tester as advertiser using a device address that differs from the IUT address in the most significant octet and a public address type.
10. Repeat steps 3–8.
11. Configure Lower Tester as advertiser using a device address that differs from the IUT address in the least significant octet and a public address type.
12. Repeat steps 3–8.
13. Configure Lower Tester as advertiser using a device address that differs from the IUT address in the most and least significant octets and a public address type.
15. Configure Lower Tester advertising channel map using the second supported advertising channel.
16. Repeat steps 2–14.
17. Configure Lower Tester advertising channel map using the third supported advertising channel.
18. Repeat steps 2–14.

• Expected Outcome

**Pass Verdict**
The test procedure completes using each supported advertising channel separately with the IUT sending a scan request to each advertiser device,
The IUT reports the advertisers with both HCI events,
The timing deviations detected for packets in active mode are within the 2 µs range around T_IFS.

• Notes
Note that neither the order nor composition with which the advertising packets are reported by the scanner is fixed.
The Controller may queue the advertising reports and some information during a scan in one HCI LE Advertising Report event.
The test procedure may be executed with filtering set to accept only the Lower Tester address or with the filtering set to white list all unknown devices. In the latter case test realization has to allow detection of other devices than the test configuration.

4.2.3.5 LL/DDI/SCN/BV-04-C [Active Scanning Device Filtering]

• Test Purpose
Tests that a scanner IUT detects, requests and reports additional information about a single advertiser according to the filtering policy and type of advertising event used.
The Lower Tester advertises using connectable undirected events on one channel at a time, expects the IUT to send the request, then when applicable responds to the IUT. The advertising packets’ payload is a sequence numbering of the packets sent.

• Reference
[3] 4.3.3, 4.4.3.2

• Initial Condition
State: Active Scanning (public address, selected scan interval, selected scan window) AND Specific White Listed (one white listed device address, one public type address, policy for scanner, black list all unknown devices)

• Test Procedure
Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.
1. Upper Tester enables active scanning with filtering policy set to ‘Ignore advertising packets from devices not in the White List Only (0x01)’ in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel.
3. Configure Lower Tester as advertiser using the white listed device address and a public address type. Scan response is also configured in the Lower Tester.
4. Lower Tester sends ADV_IND packets each advertising event using the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20) or until step 4 executes.
5. Lower Tester receives an SCAN_REQ packet T_IFS after any of the ADV_IND packets and sends an SCAN_RSP packet to the IUT T_IFS after the SCAN_REQ.

6. Interleave with step 3: Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the advertising packets.


8. Configure Lower Tester as advertiser using a device address other than the white listed device address and a public address type. The device address shall differ from the IUT address in the least significant octet. Scan response is also configured in the Lower Tester.

9. Lower Tester sends ADV_IND packets each advertising event using the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20).

10. Expect no SCAN_REQ in response after T_IFS to any of the packets.

11. Configure Lower Tester as advertiser using a device address other than the white listed device address and a public address type. The device address shall differ from the IUT address in the most and least significant octets. Scan response is also configured in the Lower Tester.

12. Repeat steps 8–9

13. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and receive an HCI_Command_Complete event in response.

14. Configure Lower Tester advertising channel map using the second supported advertising channel.

15. Repeat steps 2–13.

16. Configure Lower Tester advertising channel map using the third supported advertising channel.

17. Repeat steps 2–13.

**Expected Outcome**

**Pass Verdict**

The test procedure completes using each supported advertising channel with the timing combination, with the IUT sending a request to the white listed devices,

The IUT reports the information in the advertising packets of the white listed devices.

4.2.3.6 LL/DDI/SCN/BV-05-C [Scanning For Advertiser Types]

**Test Purpose**

Tests that a scanner IUT detects devices using different types advertising events when scanning actively and not filtering devices.

The Lower Tester sends advertising packets using several device addresses on multiple channels, using different responses and types of advertising events and observes the IUT reporting the advertisers. Some of the advertising packets include data.

**Reference**

[3] 4.4.3.1, 4.4.3.2

**Initial Condition**


State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner)
- **Test Procedure**

Execute the test advertising using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).

1. **Upper Tester** enables active scanning in the IUT.
2. Configure **Lower Tester** advertising channel map using the first supported advertising channel, a public device address that differs from the IUT address in the most significant octet and non-connectable advertising packet type.
3. **Lower Tester** sends ADV_NONCONN_IND packets each advertising event. Repeat up to a number of scan intervals (20) or step 4 is executed.
4. **Upper Tester** receives at least one HCI_LE_Advertising_Report Event containing the information used in the ADV_NONCONN_IND packets.

**Figure 4.95: LL/DDI/SCN/BV-05-C [Scanning for Advertising types]**
5. Configure Lower Tester advertising channel map using the second supported advertising channel, a public device address that differs from the IUT address in the least significant octet and undirected advertising packet type.

6. Lower Tester sends ADV_SCAN_IND packets in each advertising event using the selected advertising interval. Repeat until step 7 or up to a number of scan intervals (30).

7. Lower Tester receives an SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet to the IUT T_IFS after the SCAN_REQ.

8. Interleave with step 6: Upper Tester receives an HCI_LE_Advertising_Report Event containing the advertising packet information.


10. Configure Lower Tester advertising channel map using the third supported advertising channel, a public device address that differs from the IUT address in the most and least significant octets and directed advertising packet type.

11. Lower Tester sends ADV_DIRECT_IND packets with the IUT device address for the initiator, each advertising event using the directed advertising packet interval. Repeat up to a number of scan intervals (15) or step 8 is executed.

12. Upper Tester receives an HCI_LE_Advertising_Report Event containing the advertising packet information.

13. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

• Expected Outcome

Pass Verdict

The test procedure completes applying each advertiser address and advertising event type.

The IUT reports the packet information for the non-connectable and directed and undirected connectable packets to which a response packet was not sent.

The IUT reports the connectable packets to which a response packet was sent.

• Notes

The test procedure may be executed with filtering set to accept only the Lower Tester address or with the filtering set to white list all unknown devices. In the latter case test realization has to allow detection of other devices than the test configuration.

4.2.3.7 LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected Events]

• Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using undirected advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester. The advertising packets' payload is a sequence numbering of the packets.

• Reference

[3] 4.4.3.1

• Initial Condition

State: Passive Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner)

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

1. Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
2. Upper Tester enables passive scanning in the IUT.
3. Lower Tester sends ADV_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
4. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in step 3 and data included in one of the advertising packets.
5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.
6. Configure Lower Tester as advertiser using the second supported advertising channel using an address that differs from the IUT address in the least significant octet.
7. Repeat steps 2–5.
8. Configure Lower Tester as advertiser using the third supported advertising channel using an address that differs from the IUT address in the most significant octet and a random address type.
9. Repeat steps 2–5.

Figure 4.96: LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected events]
• Expected Outcome

**Pass Verdict**

The test procedure completes using each supported advertising channel separately, with the IUT recognizing the advertising packet type.

4.2.3.8 LL/DDI/SCN/BV-11-C [Passive Scanning: Directed Events]

• Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using directed advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester.

• Reference

[3] 4.4.3.1

• Initial Condition


State: Device Address Set (supported type of address, any address) AND Passive Scanning (selected scan interval, selected scan window) AND Specific White Listed (policy for scanner)

• Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.
1. Configure Lower Tester as advertiser using a first supported advertising channel, directed advertising with the TargetA set to the IUT’s address, and the AdvA set to a common public address in the IUT’s White List with a value that could be a valid random address (most significant bit is set to 0 or following bit is set to 1).

2. Upper Tester enables passive scanning in the IUT.

3. Lower Tester sends ADV_DIRECT_IND packets, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.

4. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in step 3.

5. Upper Tester sends an HCI_LE_Set.Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

6. Configure Lower Tester as advertiser using the second supported advertising channel, directed advertising with the TargetA set to the IUT’s address, and the AdvA set to a public address not in the IUT’s White List.

7. Repeat steps 2–5, except that in step 4, the Upper Tester does not receive an HCI_LE_Advertising_Report event.

8. Configure Lower Tester as advertiser using the third supported advertising channel, directed advertising with the TargetA set to the IUT’s address, and the AdvA set to a random address with the same 48-bit value as used in step 1.
9. Repeat steps 2–5, except that in step 4, the Upper Tester does not receive an HCI_LE_Advertising_Report event.

• Expected Outcome
  
  Pass Verdict
  
The test procedure completes using a specific advertising channel with the timing combination.

  The IUT reports advertising for the device address matching the white listed address both for the address and type with an HCI event.

  The IUT does not report the device addresses mismatching the white listed address with an HCI event.

  The IUT does not report advertising for the device address matching the white listed address for the address but mismatching the type with an HCI event.

4.2.3.9 LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable Events]

• Test Purpose
  
  Tests that a scanner IUT detects and reports advertising packets correctly.

  The Lower Tester advertises using discoverable undirected advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester. The advertising packets' payload is a sequence numbering of the packets.

• Reference
  
  [3] 4.4.3.1

• Initial Condition
  

  State: Device Address Set (supported type of address, any address) AND Passive Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner)

• Test Procedure
  
  Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.
Figure 4.98: LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable events]

1. Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
2. Upper Tester enables passive scanning in the IUT.
3. Lower Tester sends ADV_SCAN_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
4. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in step 3 and data included in one of the advertising packets.
5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.
6. Configure Lower Tester as advertiser using the second supported advertising channel and a public device address that differs from the IUT address in the least significant octet.
7. Repeat steps 2–5.
8. Configure Lower Tester as advertiser using the third supported advertising channel using an address that differs from the IUT address in the most significant octet and a random address type.
9. Repeat steps 2–5.

• Expected Outcome

  **Pass Verdict**
  
  The test procedure completes using a specific advertising channel with the timing combination.
  
  The IUT reports the device address matching the white listed both for the address and type with an HCI event,
  
  The IUT does not report the device address matching the white listed for the address and but mismatching the type with an HCI event,
  
  The IUT does not report the device addresses mismatching the white listed address with an HCI event.

**4.2.3.10 LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]**

• Test Purpose

  Verify the IUT when doing passive scanning and using the Resolving List reports advertising from the Lower Tester. The Lower Tester is doing non-connectable advertising and uses a resolvable private address for the AdvA field, i.e. the Lower Tester has distributed its own IRK.
  
  Verify that the IUT, when doing passive scanning and address resolution, is disabled and does not resolve the advertising address of the Lower Tester.

• Reference

  [3] 4.4.3.1

• Initial Condition


  State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertising packets (0x00)).

  The IUT is using a resolvable private address (0x02 or 0x03).
Figure 4.99: LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]
1. The Upper Tester populates the IUT resolving list with the peer IRK and identity address.
2. The Upper Tester enables passive scanning in the IUT.
3. Configure the Lower Tester to start advertising. The Lower Tester uses a resolvable private address in the AdvA field.
4. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
5. The Upper Tester receives at least one HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester. The address in the report is resolved by the IUT using the distributed IRK.
6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.
7. The Upper Tester disables address resolution.
8. The Upper Tester enables passive scanning in the IUT.
9. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
10. The IUT does not resolve the Lower Tester's address and reports it unresolved (as received in the advertising PDU) in the advertising report events to the Upper Tester.
11. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.

- **Expected Outcome**

  **Pass Verdict**

  The IUT receives and reports advertising with a resolvable private address in the AdvA field from the Lower Tester.

  The IUT does not resolve the address of the Lower Tester and reports it unresolved in the advertising report events to the Upper Tester when address resolution is disabled.

### 4.2.3.11 LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning: Directed Events to an address different from the scanner's address]

- **Test Purpose**

  Verify that a scanner IUT detects and reports directed advertising packets correctly, when the Upper Tester has set the scan filter policy to 0x02. The Upper Tester sets the own address to a resolvable private address, the Lower Tester sends connectable directed advertising packets to the IUT with InitA set to a resolvable private address that is different from the address that the Upper Tester has provided to the IUT. The IUT reports the directed advertising packets to the Upper Tester.

- **Reference**

  [3] 4.4.3.1, 4.3.3

- **Initial Condition**


  State: Passive Scanning (selected scan interval, selected scan window) AND extended filtering policy to process all advertising packets (0x02)

  The IUT is using a resolvable private address (0x02 or 0x03).

  The Lower Tester is also using a resolvable private address.
• **Test Procedure**

1. The Upper Tester sets a resolvable private address for the IUT to use.
2. The Upper Tester enables passive scanning using filter policy 0x02 in the IUT.
3. Configure the Lower Tester to start advertising. The Lower Tester uses a resolvable private address type in the AdvA field. The InitA field also contains a resolvable private address, which does not match the address set by the Upper Tester in the IUT.
4. The Lower Tester sends an ADV_DIRECT_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.

---

Figure 4.100: LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning: Directed Events to an address different from the scanner’s address]
5. The Upper Tester receives at least one HCI_LE_Direct_Advertising_Report reporting the advertising packets sent by the Lower Tester.
6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.

- Expected Outcome
  
  Pass Verdict
  
The IUT receives directed advertising with a resolvable private address different from the IUT address in the InitA field from the Lower Tester. The advertising packets are reported to the Upper Tester.

4.2.3.12 LL/DDI/SCN/BV-15-C [Network Privacy – Active Scanning, no Local IRK, no Peer IRK]

- Test Purpose
  
  Verify that the IUT when doing active scanning reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a non-resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising. The Lower Tester has not distributed its own IRK.

- Reference
  
  [3] 4.4.3.2, 6.3, 6.5

- Initial Condition
  
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map
  
  State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))
• **Test Procedure**

1. Configure the Lower Tester to start advertising.
2. Configure the Lower Tester as an advertiser using a random static address in the AdvA field.
3. Configure the IUT with a non-resolvable private address using the `HCI_LE_Set_Random_Address` command.
4. Upper Tester enables address resolution with all zero IRKs and random static peer address and random peer address type.
5. Upper Tester enables active scanning with filtering policy set to ‘Accept all advertising packets (0x00)’ in the IUT. The Own_Address_Type is set to 0x03.

6. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.

7. Lower Tester receives a SCAN_REQ packet T_IFS after any of the ADV_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a non-resolvable private address.

8. Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.

9. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

• Expected Outcome

Pass Verdict

The IUT receives undirected advertising using a private address in the AdvA field from the Lower Tester.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a non-resolvable private address in the ScanA field.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.13  LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK]

• Test Purpose

Verify that the IUT, when doing active scanning and using the Resolving List, reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a public or static random address for the AdvA field. The Lower Tester has not distributed its own IRK.

• Reference

[3] 4.4.3.2

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))
• **Test Procedure**

![Diagram](image-url)

**Figure 4.102: LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK]**
1. Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a random static address.
2. Configure the Lower Tester as an advertiser using its device identity address in the AdvA field.
3. Upper Tester adds the peer device to the resolving list with a valid local IRK and no peer IRK.
4. Upper Tester enables active scanning with filtering policy set to ‘Accept all advertising packets (0x00)’ in the IUT.
5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 6 completes.
6. Lower Tester receives a SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a resolvable private address.
7. Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
8. Upper Tester restarts the scanning session on the IUT.
9. Wait for address refresh timeout starting from the first SCAN_REQ received and repeat steps 5 and 6 to verify that the ScanA field in the SCAN_REQ packet has now changed to a new resolvable private address.
10. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

• Expected Outcome

Pass Verdict
The IUT receives undirected advertising using the Lower Tester's Identity Address in the AdvA field from the Lower Tester.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The ScanA field should contain a new resolvable private address after the address refresh timeout.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.14  LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK]

• Test Purpose

Verify that the IUT, when doing active scanning and using the Resolving List reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a non-resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a resolvable private address for the AdvA field. The Lower Tester has distributed its own IRK.

• Reference

[3] 4.4.3.2

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))
• Test Procedure

1. Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a non-resolvable private address.
2. Configure the Lower Tester advertising channel map using the first supported advertising channel.

Figure 4.103: LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK]
3. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.
4. Upper Tester adds peer identity and IRK information to resolving list.
5. Upper Tester enables active scanning with filtering policy set to ‘Accept all advertising packets (0x00)’ in the IUT.
6. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
7. Lower Tester receives a SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a non-resolvable private address.
8. Upper Tester receives an HCI_LE_Advertising_Report event containing the ADV_SCAN_IND response information sent in step 7.

• Expected Outcome
  
  **Pass Verdict**
  
  The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester.

  The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a non-resolvable private address in the ScanA field.

4.2.3.15 LL/DDI/SCN/BV-18-C [Network Privacy – Active Scanning, Local IRK, Peer IRK]

• Test Purpose
  
  Verify that the IUT when doing active scanning and using the Resolving List reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a resolvable private address for the AdvA field i.e. the IUT and Lower Tester has distributed its Device Identities.

• Reference
  
  [3] 4.4.3.2

• Initial Condition
  
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

  State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))
**Test Procedure**

1. Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a random static address.
2. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.
3. Upper Tester adds peer device identity and local IRK information to resolving list.
4. Upper Tester enables active scanning with filtering policy set to ‘Accept all advertising packets (0x00)’ in the IUT.
5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
6. Lower Tester receives a SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. The ScanA field in the SCAN_REQ packet shall use the same resolvable private address.

7. Lower Tester sends a SCAN_RSP packet T_IFS after the SCAN_REQ packet. The AdvA field in the SCAN_RSP packet should use the resolvable private address that was used in the SCAN_REQ packet.

8. Interleave with step 6: Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.


- Expected Outcome

  **Pass Verdict**

  The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester. The IUT successfully resolves the address.

  The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The Lower Tester successfully resolves the address.

  IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

### 4.2.3.16 Extended Scanning, Passive

- **Test Purpose**

  Tests that a scanner IUT detects and reports advertising packets received correctly, including legacy and extended PDUs. The Lower Tester advertises using non-connectable and non-scannable advertising events on the primary and secondary advertising channels and expects the IUT to report the advertising to the Upper Tester. Both directed and undirected advertising events are tested, with and without data.

- **Reference**

  [10] 4.4.3.1

- **Initial Condition**


  State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

  The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IXIT parameter.

- **Test Procedure**

  Execute the test procedure using the minimum advertising interval and continuous passive scanning.
For each round as specified in Table 4.24 based on Table 4.23:

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters_Command to the IUT. The Scanning PHYs parameter shall be set as specified in Table 4.24, Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates and Period shall be set to zero. The Duration parameter shall be set to the value specified in Table 4.23 for this round.

3. The Lower Tester begins advertising using the PDU Type specified in Table 4.23 for this round. If AUX_ADV_IND is included in the round, the ADV_EXT_IND shall include an AuxPtr that refers to the AUX_ADV_IND, and all fields specified should be included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where “LT” equals the Lower Tester address. If InitA/TargetA is specified the appropriate PDU shall include the field, where “IUT” equals the IUT address and “Not IUT” equals a random address other than the IUT address.
AdvData is specified the PDU shall include the field populated with random octets of the specified count. If the AdvData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data. Each PDU except the last shall contain as much AdvData as can fit. If Duration is set to 0x0000, repeat for at least 20 advertising intervals, otherwise repeat until the end of the round.

4. For undirected advertisements or advertisements directed at the IUT, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report events from the IUT with an advertising event type matching the type sent in step 3 and the Primary_PHY set as specified in Table 4.24, and if the advertisements used extended PDUs, the Secondary_PHY shall be set as specified in Table 4.24. If AdvData was included in the advertisement, the Upper Tester receives the data included in one or more of the advertising packets. If the advertisement was directed at the IUT, the Upper Tester receives the Direct Address Type and Direct Address used to direct the advertisement at the IUT.

5. If the Duration was set to 0x0000 (No Scanning Duration), repeat step 4 until a number of advertising reports (10) have been generated. Each time the Upper Tester receives a report, the Lower Tester shall change the AdvData, if any. If the round uses extended advertising PDUs, it shall also change the DID sub-field of the ADI field to a new value. Otherwise repeat step 4 until the amount of time specified for Duration has elapsed. Afterwards, the Upper Tester receives an HCI_Scan_Timeout event from the IUT. Skip step 6.

6. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

7. Repeat steps 1–5 for each Round shown in Table 4.23.

<table>
<thead>
<tr>
<th>Round</th>
<th>LE Set Extended Scan Enable (Step 2)</th>
<th>Lower Tester Advertising PDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration</td>
<td>PDU Type (AdvMode)</td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>0x0000</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>3</td>
<td>0x0000</td>
<td>ADV_DIRECT_IND</td>
</tr>
<tr>
<td>4</td>
<td>0x0000</td>
<td>ADV_DIRECT_IND</td>
</tr>
<tr>
<td>5</td>
<td>0x0000</td>
<td>ADV_NONCONN_IND</td>
</tr>
<tr>
<td>6</td>
<td>0x0000</td>
<td>ADV_NONCONN_IND</td>
</tr>
<tr>
<td>7</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td>8</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td>9</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td>10</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
</tbody>
</table>
### Table 4.23: PDU payload contents for each case variation.

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Round</th>
<th>LE Set Extended Scan Enable (Step 2)</th>
<th>Lower Tester Advertising PDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration</td>
<td>PDU Type (AdvMode)</td>
</tr>
<tr>
<td>11</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td>12</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td>13</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td>14</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
</tr>
<tr>
<td>15</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
</tr>
<tr>
<td>16</td>
<td>0x01F4 (5 s)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td>17</td>
<td>0x0000</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
</tbody>
</table>

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
<th>Scanning PHY</th>
<th>Primary Advertising PHY</th>
<th>Secondary Advertising PHY</th>
<th>Rounds to be Executed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.2.3.16.1 LL/DDI/SCN/BV-19-C [Extended Scanning, Passive – LE 1M PHY]</strong></td>
<td></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
<td>1–17</td>
</tr>
<tr>
<td><strong>4.2.3.16.2 LL/DDI/SCN/BV-42-C [Extended Scanning, Passive – LE 2M PHY]</strong></td>
<td></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (LE 2M PHY)</td>
<td>10–17</td>
</tr>
</tbody>
</table>
### Test Case

<table>
<thead>
<tr>
<th>Scanning PHY</th>
<th>Primary Advertising PHY</th>
<th>Secondary Advertising PHY</th>
<th>Rounds to be Executed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04 (LE Coded PHY)</td>
<td>0x03 (LE Coded PHY)</td>
<td>0x03 (LE Coded PHY)</td>
<td>10–17</td>
</tr>
</tbody>
</table>

**Table 4.24: Extended Scanning, Passive Test Cases**

- **Expected Outcome**

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- For undirected advertisements or advertisements directed at the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events with an advertising event type matching the type sent in step 3, the Address and Address_Type fields match the Lower Tester address, the Primary_PHY set correctly, and if the advertisements used extended PDUs, the Secondary_PHY shall be set correctly.

- If AdvData was included in the advertisement, the Upper Tester receives the correct data in the Data field, in one or more HCI_LE_Extended_Advertising_Report events. Either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type[i] specifying "Complete". In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type[i] specifying "Incomplete, data truncated, no more to come". In either case the preceding events shall have an Event_Type[i] specifying "Incomplete, more data to come". The IUT shall not truncate if the data length is no more than Scan_Max_Data.

- If the advertisement was directed at the IUT, the Upper Tester receives the Direct Address Type and Direct Address used to direct the advertisement at the IUT.

- For directed advertisements not directed to the IUT, the IUT does NOT generate an HCI_LE_Extended_Advertising_Report event or receive the data.

- If a nonzero Duration was specified, the IUT generates an LE Scan Timeout event after the duration has elapsed and stops scanning.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 4, the Advertising_SID value in the event is the value in the ADI field of the advertising transmitted in step 3, or 0xFF if that advertising does not include an ADI field.

### 4.2.3.17 Extended Scanning, Active

- **Test Purpose**

Tests that a scanner IUT detects and requests additional information from advertisements received and reports the results from the Controller. The Lower Tester advertises using scannable extended advertising events on one channel at a time and expects the IUT to report the advertising to the Upper Tester. Both directed and undirected advertising events are tested.

- **Reference**

[10] 4.4.3.2
• Initial Condition


State: Active Scanning (Active, selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IUT parameter.

• Test Procedure

Execute the test procedure using the minimum advertising interval and continuous active scanning.

1. For each round as specified in Table 4.26 based on Table 4.25, if ScanData Length is less than or equal to the “Scan Max Data” then perform steps 2–8 and otherwise omit this round.

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set as specified in Table 4.26, Scan_Type[0] set to 0x01
(Active Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010.
Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy
shall be set to 0x00 (Accept All).

3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable
scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

4. The Lower Tester begins advertising on the channel as specified in Table 4.26 using the PDU
Type specified in Table 4.25 for this round. If ADV_ADV_IND is included in the round, the
ADV_EXT_IND shall include an AuxPtr that refers to the AUX_ADV_IND on the PHY as specified
in Table 4.26, and all fields specified should be included with the AUX_ADV_IND only. If AdvA is
specified the appropriate PDU shall include the field, where “LT” equals the Lower Tester
address. If TargetA is specified the appropriate PDU shall include the field, where “IUT” equals
the IUT address and “Not IUT” equals a random address other than the IUT address. Repeat for
at least 20 advertising intervals or until step 5 occurs.

5. For undirected advertisements or advertisements directed at the IUT, the Lower Tester receives
either a SCAN_REQ (if advertising with legacy PDUs) or an AUX_SCAN_REQ (if advertising with
extended PDUs) on the appropriate advertising channel. The ScanA field shall be set to the IUT’s
address and the AdvA address set to the Lower Tester’s address. The Upper Tester receives one
or more HCI_LE_Extended_Advertising_Report events from the IUT with an Event_Type where
bit 3 (Scan response) is not set. If the advertisement was directed at the IUT, the Upper Tester
receives the Direct Address Type and Direct Address used to direct the advertisement at the IUT.
If the advertisements were not directed at the IUT, skip to step 8.

6. If the IUT sent a SCAN_REQ in step 5, the Lower Tester responds with an SCAN_RSP packet to
the IUT T_IFS after the end of the SCAN_REQ PDU. If ScanData is specified, the SCAN_RSP
PDU shall include the field populated with random octets from 1 to 255, of the specified count.
OR
If the IUT sent an AUX_SCAN_REQ in step 5, the Lower Tester responds with an
AUX_SCAN_RSP packet to the IUT T_IFS after the end of the AUX_SCAN_REQ PDU with an
AdvMode of 00b. If ScanData is specified, the AUX_SCAN_RSP PDU shall include the AdvData
field populated with random octets from 1 to 255, of the specified count. If the ScanData is
greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send
one or more AUX_CHAIN_IND PDUs containing the remaining data. Each PDU except the last
shall contain as much AdvData as can fit.

7. If the Lower Tester sent a scan response in step 6, the Upper Tester receives one or more
HCI_LE_Extended_Advertising_Report events from the IUT with an Event_Type where bit 3
(Scan response) is set. If ScanData was included in the response, the Upper Tester receives the
data included in one of the advertising packets.

8. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives
an HCI_Command_Complete event in response.

<table>
<thead>
<tr>
<th>Round</th>
<th>Lower Tester Advertising PDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PDU Type (AdvMode)</td>
</tr>
<tr>
<td>1</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>3</td>
<td>ADV_SCAN_IND</td>
</tr>
<tr>
<td>4</td>
<td>ADV_SCAN_IND</td>
</tr>
</tbody>
</table>
### Lower Tester Advertising PDUs

<table>
<thead>
<tr>
<th>Round</th>
<th>PDU Type (AdvMode)</th>
<th>AdvA</th>
<th>TargetA</th>
<th>ScanData</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>IUT</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>Not IUT</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>-</td>
<td>191</td>
</tr>
<tr>
<td>9</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>-</td>
<td>382</td>
</tr>
<tr>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>ADV_EXT_IND (10b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AUX_ADV_IND (10b)</td>
<td>LT</td>
<td>-</td>
<td>Scan_Max_Data</td>
</tr>
<tr>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 4.25: PDU payload contents for each case variation.*

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scanning PHY</strong></td>
<td><strong>Primary Advertising PHY</strong></td>
<td><strong>Secondary Advertising PHY</strong></td>
</tr>
<tr>
<td><strong>4.2.3.17.1 LL/DDI/SCN/ BV-20-C [Extended Scanning, Active – LE 1M PHY, Core 5.0]</strong></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.2.3.17.2 LL/DDI/SCN/ BV-44-C [Extended Scanning, Active – LE 2M Phy, Core 5.0]</strong></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (LE 2M PHY)</td>
</tr>
</tbody>
</table>
### Test Case | PHYs | Advertising SID
---|---|---
| Scanning PHY | Primary Advertising PHY | Secondary Advertising PHY | Rounds to be Executed |
| | | | |
| **4.2.3.17.3** LL/DDI/SCN/BV-45-C [Extended Scanning, Active – LE Coded PHY, Core 5.0] | 0x04 (LE Coded PHY) | 0x03 (LE Coded PHY) | 0x03 (LE Coded PHY) | 5–8 |
| | 0xFF or the advertising SID from the AUX_ADV_IND PDU. |
| **4.2.3.17.4** LL/DDI/SCN/BV-64-C [Extended Scanning, Active – LE 1M PHY, Core 5.1] | 0x01 (LE 1M PHY) | 0x01 (LE 1M PHY) | 0x01 (LE 1M PHY) | 1–10 |
| | Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU. |
| **4.2.3.17.5** LL/DDI/SCN/BV-65-C [Extended Scanning, Active – LE 2M Phy, Core 5.1] | 0x01 (LE 1M PHY) | 0x01 (LE 1M PHY) | 0x02 (LE 2M PHY) | 5–8 |
| | Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU. |
| **4.2.3.17.6** LL/DDI/SCN/BV-66-C [Extended Scanning, Active – LE Coded PHY, Core 5.1] | 0x04 (LE Coded PHY) | 0x03 (LE Coded PHY) | 0x03 (LE Coded PHY) | 5–8 |
| | Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU. |

Table 4.26: Extended Scanning, Active Test Cases

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:

  - For undirected advertisements or advertisements directed at the IUT, the IUT sends a scan request PDU to the LT.
  
  - For scan responses received by the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events with an advertising event type matching the type sent in step 4. If ScanData was included in the advertisement, either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type[i] specifying "Complete". In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type[i] specifying "Incomplete, data truncated, no more to come". In either case the preceding events shall have an Event_Type[i] specifying "Incomplete, more data to come". The IUT shall not truncate if the data length is no more than Scan_Max_Data.
- If the advertisement was directed at the IUT, the Upper Tester receives the Direct Address Type and Direct Address used to direct the advertisement at the IUT.

- For directed advertisements not directed to the IUT, the IUT does NOT send a scan request or generate an HCI_LE_Extended_Advertising_Report event.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value in the ADI field of the advertising transmitted in step 4, or 0xFF if that advertising does not include an ADI field.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in response to an AUX_SCAN_RSP packet, the Advertising_SID value in the event is the value specified in Table 4.26.

**4.2.3.18 Extended Scanning, Periodic Advertising Reception**

• **Test Purpose**

Tests that a scanner IUT can locate and receive periodic advertising events and reports the results from the Controller. The Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising and expects the IUT to report the periodic advertising data by the Upper Tester. Confirms the IUT receives periodic advertisement PDUs sent using the Channel Selection Algorithm #2.

• **Reference**

[10] 4.4.2.13.1, 4.4.3.4

• **Initial Condition**


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IXIT parameter.

• **Test Procedure**

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.
For each round as specified in Table 4.28 based on Table 4.27:

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set as specified in Table 4.28, Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

3. The Lower Tester begins advertising using ADV_EXT_IND and AUX_ADV_IND PDUs using the values as specified in Table 4.28. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to
the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in step 3. If the Periodic Data Length column in Table 4.27 for this Round is non-zero, the AUX_SYNC_IND PDUs shall contain an AdvData field where the first 2 octets contain a 16-bit counter which is initially 0 and is incremented at each periodic event, and the remainder contain random octets, up to Periodic Data Length in Table 4.27 or the maximum that can fit in the PDU. If all the data cannot fit in a single AUX_SYNC_IND PDU, the Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs containing the remaining data. Each PDU except the last shall contain as much AdvData as can fit. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements. Options shall be set to 0x00 (Don’t Use List). Advertising_SID shall be set to the Advertising_SID from step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester’s address. Skip shall be set to the Skip value in Table 4.27. Sync_Timeout shall be set to (Skip + 3) x Periodic_Advertising_Interval from step 5. Sync_CTE_Type shall be set to 0x00. The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. If AdvData is being advertised, the Upper Tester receives the data included in the advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to 0xFF.

9. Repeat step 8 until at least 30 HCI_LE_Periodic_Advertising_Report events if current round is round 7, and 5 events for other rounds have been received.

10. The Lower Tester ceases extended advertising but continues periodic advertising.

11. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report events from the IUT as in step 8.

12. Repeat step 11 until at least 30 events have been received if current round is round 7, and until at least 5 events have been received for other rounds.

13. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

14. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

15. The Lower Tester ceases periodic advertising.

16. Repeat steps 1–15 for each Round shown in Table 4.27.
### Table 4.27: PDU payload contents and Skip value for each case variation.

<table>
<thead>
<tr>
<th>Round</th>
<th>Lower Tester Periodic Advertisements (Step 4)</th>
<th>HCI_LE_Periodic_Advertising_Create_Sync (Steps 4 and 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Periodic Data Length</td>
<td>Skip</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>0x0000</td>
</tr>
<tr>
<td>2</td>
<td>31 octets</td>
<td>0x0000</td>
</tr>
<tr>
<td>3</td>
<td>191 octets</td>
<td>0x0000</td>
</tr>
<tr>
<td>4</td>
<td>382 octets</td>
<td>0x0000</td>
</tr>
<tr>
<td>5</td>
<td>Scan_Max_Data</td>
<td>0x0000</td>
</tr>
<tr>
<td>6</td>
<td>Scan_Max_Data</td>
<td>0x0001</td>
</tr>
<tr>
<td>7</td>
<td>Scan_Max_Data</td>
<td>0x00F9</td>
</tr>
<tr>
<td>8</td>
<td>Scan_Max_Data</td>
<td>0x01F3</td>
</tr>
<tr>
<td>9</td>
<td>31 octets</td>
<td>0x000F</td>
</tr>
</tbody>
</table>

### Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Scanning PHY</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2.3.18.1 LL/DDI/SCN/BV-21-C</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td></td>
<td>LL/DDI/SCN/BV-21-C</td>
</tr>
<tr>
<td></td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td></td>
<td>LL/DDI/SCN/BV-46-C</td>
</tr>
<tr>
<td></td>
<td>0x04 (LE Coded PHY)</td>
</tr>
<tr>
<td></td>
<td>LL/DDI/SCN/BV-47-C</td>
</tr>
</tbody>
</table>

### Table 4.28: Extended Scanning, Periodic Advertising Reception Test Cases
• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements using the Channel Selection Algorithm #2.

- If AdvData was included in the advertisement, the Upper Tester receives the correct data in the Data field, in one or more HCI_LE_Periodic_Advertising_Report events. Either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type specifying "Complete". In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type specifying "Incomplete, data truncated, no more to come". In either case the preceding events shall have an Event_Type specifying "Incomplete, more data to come". The IUT shall not truncate if the data length is no more than Scan_Max_Data.

- The IUT maintains synchronization with the Lower Tester periodic advertisement PDUs even after the Lower Tester ceases extended advertising.

- The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.

- Where the periodic advertising data contains a counter, the difference between the counter value in consecutive reports is less than or equal to Skip + 1 in at least 95 percent (rounded down) of the cases.

- In the HCI_LE_EXTENDED_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value sent by the Lower Tester in step 3.

4.2.3.19 LL/DDI/SCN/BV-23-C [Extended Scanning, Multiple Sets, Passive, Multiple PHYs (All Supported PHYs)]

• Test Purpose

Tests that a scanner IUT detects and reports advertising packets received on all supported PHYs correctly. Using all PHYs, the Lower Tester advertises using non-connectable and non-scannable extended advertising events on one channel at a time and expects the IUT to report the advertising to the Upper Tester.

• Reference

[10] 4.4.3.1

• Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IXIT parameter.

• Test Procedure

Execute the test procedure using the minimum advertising interval and continuous passive scanning on all supported PHYs.
For each round in Table 4.29:

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters_Command to the IUT. The Scanning_PHYs parameter shall be set to all supported PHYs. For each supported PHY i, Scan_Type[i] shall be set to 0x00 (Passive Scanning), Scan_Interval[i] set to (0x0010 * number of bits set in Scanning_PHY), and Scan_Window[i] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

3. The Lower Tester begins advertising six advertising sets; the advertising_set ID shall be different for each set. The first two sets shall use the LE 1M PHY for both primary and secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary and secondary advertising channels, and the last two sets shall use the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel. Each set shall use ADV_EXT_IND PDUs as specified in Table 4.29 for this round with the AuxPtr field referencing the AUX_ADV_IND with all fields specified included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where “LT” equals the Lower Tester address. If AdvData is specified the PDU shall include the field populated with random octets of the specified count. If the AdvData is greater in length than will fit in one PDU, the Lower Tester
shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data. Each PDU except the last shall contain as much AdvData as can fit. Repeat for at least 20 advertising intervals or until step 4 occurs.

4. For each set of advertisements using a PHY the IUT supports, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report events from the IUT with an advertising event type matching the type sent in step 3 and the Primary_PHY and Secondary_PHY set to the PHYs used for the advertising set. If AdvData was included in the advertisement, the Upper Tester receives the data included in one or more of the advertising packets.

5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

<table>
<thead>
<tr>
<th>Round</th>
<th>Advertising Sets</th>
<th>Lower Tester Advertising PDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PDU Type (AdvMode)</td>
</tr>
<tr>
<td>1</td>
<td>1 &amp; 2 (LE 1M, LE 1M)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>3 &amp; 4 (LE Coded, LE Coded)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>5 &amp; 6 (LE 1M, LE 2M)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td>2</td>
<td>1 &amp; 2 (LE 1M, LE 1M)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>3 &amp; 4 (LE Coded, LE Coded)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>5 &amp; 6 (LE 1M, LE 2M)</td>
<td>ADV_EXT_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (00b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_CHAIN_IND (00b)</td>
</tr>
</tbody>
</table>

Table 4.29: PDU payload contents for each case variation

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:

  - For advertisements using PHYs the IUT does not support, the IUT does NOT generate an HCI_LE_Extended_Advertising_Report event or receive the data.
  
  - For advertisements using a PHY the IUT supports, the IUT generates one or more HCI_LE_Extended_Advertising_Report events with an advertising event type and advertising set ID matching the type and set ID sent in step 3 and the Primary_PHY and Secondary_PHY set to...
the PHYs used for the advertising set. Either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type[i] specifying "Complete". In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type[i] specifying "Incomplete, data truncated, no more to come". In either case the preceding events shall have an Event_Type[i] specifying "Incomplete, more data to come". The IUT shall not truncate if the data length is no more than Scan_Max_Data.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 4, the Advertising_SID value in the event is the value transmitted by the Lower Tester in step 3.

4.2.3.20 Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)

- Test Purpose
  Tests that a scanner IUT detects and requests additional information from advertising packets received on all supported PHYs, where the scan requests are sent on the secondary channel, and reports the results from the Controller. Using all PHYs, the Lower Tester advertises using scanable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel and expects the IUT to report the advertising to the Upper Tester.

- Reference
  [10] 4.4.3.2

- Initial Condition
  State: Active Scanning (Active, selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).
  The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IXIT parameter.

- Test Procedure
  Execute the test procedure using the minimum advertising interval and continuous active scanning on all supported PHYs.
For each round in Table 4.30:

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters_Command to the IUT. The Scanning_PHYs parameter shall be set to all supported PHYs. For each supported PHY i, Scan_Type[i] shall be set to 0x01 (Active Scanning), Scan_Interval[i] set to (0x0010 * number of bits set in Scanning_PHY), and Scan_Window[i] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

3. The Lower Tester begins advertising six advertising sets; the advertising_set ID shall be different for each set. The first two sets shall use the LE 1M PHY for both primary and secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary and secondary advertising channels, and the last two sets shall use the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel. Each set shall use ADV_EXT_IND PDUs as specified in Table 4.30 for this round with the AuxPtr field.
referencing the AUX_ADV_IND with all fields specified included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where “LT” equals the Lower Tester address. Repeat for at least 20 advertising intervals or until step 4 occurs.

4. For each set of advertisements using a PHY the IUT supports, the Lower Tester receives an AUX_SCAN_REQ on the appropriate advertising channel using the appropriate PHY. The ScanA field shall be set to the IUT’s address and the AdvA address set to the Lower Tester’s address.

5. The Lower Tester responds to each AUX_SCAN_REQ PDU with an AUX_SCAN_RSP packet to the IUT T_IFS after the end of the AUX_SCAN_REQ PDU with an AdvMode of 00b. The AUX_SCAN_RSP PDU shall include the AdvData field populated with random octets from 1 to 255, of the specified count, except that the first octet shall be the advertising_set ID for the relevant advertising set. Although Optional, the AdvDataInfo field shall not be present in the AUX_SCAN_RSP PDU. If the ScanRspData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data. Each PDU except the last shall contain as much AdvData as can fit.

6. For each scan response the Lower Tester sent, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report events from the IUT with an Event_Type where bit 3 (Scan response) is set. If ScanData was included in the response, the Upper Tester receives the data included in one or more of the advertising packets.

7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

<table>
<thead>
<tr>
<th>Round</th>
<th>Advertising Sets</th>
<th>Lower Tester Advertising PDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PDU Type (AdvMode)</td>
</tr>
<tr>
<td>1</td>
<td>1 &amp; 2 (LE 1M, LE 1M)</td>
<td>ADV_EXT_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 &amp; 4 (LE Coded, LE Coded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADV_EXT_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 &amp; 6 (LE 1M, LE 2M)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
<tr>
<td>2</td>
<td>1 &amp; 2 (LE 1M, LE 1M)</td>
<td>ADV_EXT_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUXCHAIN_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>3 &amp; 4 (LE Coded, LE Coded)</td>
<td>ADV_EXT_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUXCHAIN_IND (00b)</td>
</tr>
<tr>
<td></td>
<td>5 &amp; 6 (LE 1M, LE 2M)</td>
<td>ADV_EXT_IND (10b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUX_ADV_IND (10b)</td>
</tr>
</tbody>
</table>
• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:
- For advertisements using PHYs the IUT does not support, the IUT does NOT generate an HCI_LE_Extended_Advertising_Report event or send a scan request.
- For advertisements using a PHY the IUT supports, for each set using that PHY, the IUT sends a scan request PDU to the LT.
- For scan responses received by the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events to the Upper Tester with an advertising event type indicating Scannable advertising and Scan response (b4b3b2b1b0 = 0b01010) and the advertising set ID matching the set ID sent in step 3. Either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type[i] specifying "Complete" (b6b5 = 0b00). In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type[i] specifying "Incomplete, data truncated, no more to come". In either case the preceding events shall have an Event_Type[i] specifying "Incomplete, more data to come" (b6b5 = 0b10). The IUT shall not truncate if the data length is no more than Scan_Max_Data.
- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 6, the Advertising_SID value in the event is the value as specified in Table 4.31.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3.20.1 LL/DDI/SCN/BV-24-C Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs), Core 5.0</td>
<td>0xFF or the Advertising SID from the first octet of the scan response data.</td>
</tr>
<tr>
<td>4.2.3.20.2 LL/DDI/SCN/BV-62-C Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs), Core 5.1</td>
<td>Advertising SID from the first octet of the scan response data.</td>
</tr>
</tbody>
</table>

Table 4.31: Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY test cases

4.2.3.21 LL/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)]

• Test Purpose

Tests that a scanner IUT can locate and receive periodic advertising events on all supported PHYs and reports the results from the Controller. Using all PHYs, the Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising and expects the IUT to report the periodic advertising data to the Upper Tester.

• Reference

[10] 4.4.2.13.1, 4.4.3.4
• Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the Scan_Max_Data IXIT parameter.

• Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester using all supported PHYs.
For each round from 1 to 5 based on Table 4.32:

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters_Command to the IUT. The Scanning_PHYs parameter shall be set to all supported PHYs. For each supported PHY i, Scan_Type[i] shall be set to 0x00 (Passive Scanning), Scan_Interval[i] set to (0x0010 * number of bits set in Scanning_PHY), and Scan_Window[i] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising six advertising sets using ADV_EXT_IND and AUX_ADV_IND PDUs. The first two sets shall use the LE 1M PHY for both primary and
secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary
and secondary advertising channels, and the last two sets shall use the LE 1M PHY for the
primary advertising channel and the LE 2M PHY for the secondary advertising channel. For each
advertising set, the ADV_EXT_IND PDUs shall include an AuxPtr that refers to the
AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall
include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the
AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test
procedure.

4. For each advertising set, the Lower Tester generates ADV_SYNC_IND PDUs on the secondary
advertising channel as specified in the SyncInfo in step 3. If the Data Length column in Table 4.32
for this Round is non-zero, the AUX_SYNC_IND PDUs shall contain an AdvData field containing
random octets from 1-255, up to Data Length in Table 4.32 or the maximum that can fit in the
PDU. If all the data cannot fit in a single AUX_SYNC_IND PDU, the Lower Tester shall include an
AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs
containing the remaining data. Each PDU except the last shall contain as much AdvData as can
fit. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

5. For each set using a PHY the IUT supports, the Upper Tester receives an
HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a
nonzero Periodic_Advertising_Interval. The first time that such an event is received for each set,
steps 6–12 shall be performed for that set while continuing to wait for other such events. Only one
instance of step 6 shall be carried out at a time but multiple instances of steps 7–12 may be
performed in parallel with each other and with the one instance of step 6.

6. a. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT
to synchronize with the Lower Tester’s periodic advertisements. Options shall be set to 0x00
(Don’t Use List). Advertising_SID shall be set to the Advertising_SID from step 5.
Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address
shall be set to the Lower Tester’s address. Sync_Timeout shall be set to 3 x
Periodic_Advertising_Interval from the event in step 5. Unused shall be set to 0x00.
b. The Upper Tester receives an HCI_Command_Status event in response.
c. The Upper Tester shall wait for an HCI_LE_Periodic_Advertising_Sync_Established event
from the IUT containing a Status of 0x00 (Success) and other fields matching the
advertisements generated by the Lower Tester.
d. If either the HCI_Command_Status event or the
HCI_LE_Periodic_Advertising_Sync_Established event indicates that the command failed
because the IUT has a limitation on the number of periodic advertisements it can synchronize
to at one time, this step shall be repeated when step 12 completes for another advertising
set.

7. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the
IUT. If AdvData is being advertised, the Upper Tester receives the data included in the
advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to
0xFF.

8. Repeat step 7 until at least 100 events have been received.

9. The Lower Tester ceases extended advertising on this advertising set but continues periodic
advertising.

10. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report events from the IUT
as in step 7.

11. Repeat step 10 until at least 100 events have been received.
12. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

13. When step 12 has completed for all sets using a PHY the IUT supports, the Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

14. The Lower Tester ceases periodic advertising on all advertising sets.

15. Repeat steps 1–14 for each Round shown in Table 4.32.

<table>
<thead>
<tr>
<th>Round</th>
<th>Lower Tester Periodic Advertisements (Step 4)</th>
<th>Periodic Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>None</td>
<td>1 None</td>
</tr>
<tr>
<td>2</td>
<td>31 octets</td>
<td>2 31 octets</td>
</tr>
<tr>
<td>3</td>
<td>191 octets</td>
<td>3 191 octets</td>
</tr>
<tr>
<td>4</td>
<td>382 octets</td>
<td>4 382 octets</td>
</tr>
<tr>
<td>5</td>
<td>Scan_Max_Data</td>
<td>5 Scan_Max_Data</td>
</tr>
</tbody>
</table>

Table 4.32: PDU payload contents for each case variation.

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:
  
  - For each set using a PHY the IUT supports, the IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements.
  
  - If AdvData was included in the advertisement, the Upper Tester receives the correct data in the Data field, in one or more HCI_LE_Extended_Advertising_Report events. Either the whole of the data shall be reported or the data shall be truncated. In the former case, the last event shall have an Event_Type[i] specifying “Complete”. In the latter case, the point of truncation shall be at the start of one of the AUX_CHAIN_IND PDUs (i.e., not within the contents of a single PDU) and the last event shall have an Event_Type[i] specifying “Incomplete, data truncated, no more to come”. In either case the preceding events shall have an Event_Type[i] specifying “Incomplete, more data to come”. The IUT shall not truncate if the data length is no more than Scan_Max_Data.
  
  - For each set using a PHY the IUT supports, the IUT maintains synchronization with the Lower Tester periodic advertisement PDUs even after the Lower Tester ceases extended advertising.
  
  - For each set using a PHY the IUT supports, the IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.
- For PHYs the IUT does not support, the IUT does not generate HCI_LE_Periodic_Advertising_Report events.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value sent by the Lower Tester in step 4.

4.2.3.22 LL/DDI/SCN/BV-26-C [Network Privacy – Passive Scanning, Peer IRK, Ignore Identity Address]

• Test Purpose

Verify the IUT when doing passive scanning and using the Resolving List does not report advertising from the Lower Tester using its device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode. The Lower Tester has distributed an IRK, but the Lower Tester is doing non-connectable advertising using its device identity address for the AdvA field.

• Reference

[3] 4.4.3.1

• Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertisement packets (0x00)).

The IUT is using a resolvable private address (0x02 or 0x03).

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.
• Test Procedure

1. The Upper Tester populates the IUT resolving list with the peer IRK and identity address.
2. The Upper Tester enables passive scanning in the IUT.
3. Configure the Lower Tester to start advertising. The Lower Tester uses its device identity address in the AdvA field.
4. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
5. The Upper Tester receives no HCI_LE_Advertising_Report events from the IUT.
6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.

• Expected Outcome

Pass Verdict

The IUT does not report advertising with the device identity address in the AdvA field from the Lower Tester, when the device identity address and an associated IRK are in the resolving list using network privacy mode.
4.2.3.23 LL/DDL/SCN/BV-28-C [Device Privacy – Passive Scanning, Peer IRK, Accept Identity Address]

• Test Purpose

Verify the IUT when doing passive scanning and using the Resolving List reports advertising from the Lower Tester using its device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode. The Lower Tester has distributed an IRK, but the Lower Tester is doing non-connectable advertising using its device identity address for the AdvA field.

• Reference

[3] 4.4.3.1

• Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertisement packets (0x00)).
• **Test Procedure**

1. The Upper Tester populates the IUT resolving list with the peer IRK and identity address and sets the entry to device privacy mode.
2. The Upper Tester enables passive scanning in the IUT.
3. Configure the Lower Tester to start advertising. The Lower Tester uses its device identity address in the AdvA field.
4. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
5. The Upper Tester receives at least one HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester. The address in the report is the Lower Tester’s identity address.

6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.

• Expected Outcome

**Pass Verdict**

The IUT receives and reports advertising with the device identity address in the AdvA field from the Lower Tester, when the device identity address and an associated IRK are in the resolving list using device privacy mode.

### 4.2.3.24 AoD Connectionless CTE Scanning

• **Test Purpose**

Tests that a scanner IUT detects and reports advertising events with the AoD Connectionless Constant Tone Extension included when advertiser is utilizing a public device address.

The Lower Tester is advertising with an AoD Connectionless Constant Tone Extension, using a public device address. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

• **Reference**

[13] 2.5.2, 2.5.4

• **Initial Condition**


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).
**Test Procedure**

*Repeat for 1 µs and 2 µs slots (where supported)*

### For each round from 1 to 3

- **ADV_EXT_IND**
- **AUX_ADV_IND** (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)
- **ADV_EXT_IND**
- **AUX_ADV_IND** (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)
- **AUX_SYNC_IND** (CTE)
- **AUX_CHAIN_IND** (One or more, CTE)
- **AUX_SYNC_IND** (CTE)
- **AUX_CHAIN_IND** (One or more, CTE)

**Figure 4.113: AoD Connectionless CTE Scanning**

For each round from 1 to 3 based on **Table 4.33**:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
</tbody>
</table>
1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel shall be set to 0x01 (LE 1M) and AUX_ADV_IND PDUs on the secondary advertising channel as specified in Table 4.34. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address, a SynchInfo field referring to the AUX_SYNC_IND PDUs and does not include the AdvData field. The Lower Tester continues advertising until directed to stop in the test procedure.
4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel as specified in Table 4.34. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to the value specified in Table 4.33, RFU set to '0', and CTEType set as specified in Table 4.34. The AUX_SYNC_IND PDUs shall not include the AdvData field. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in Table 4.33. While transmitting the Constant Tone Extension field, the Lower Tester switches antennas and slot durations as specified in Table 4.34. The Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs.
5. The Lower Tester generates one or more AUX_CHAIN_IND PDUs on the secondary advertising channel as specified in Table 4.34. The AUX_CHAIN_IND PDUs shall not include the AdvData field. Each packet containing an AUX_CHAIN_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in Table 4.33. While transmitting the Constant Tone Extension field, the Lower Tester switches antennas using slot durations as specified in Table 4.34. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
6. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.
7. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from step 6. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 6. CTE_Type shall be set as specified in Table 4.34. The Upper Tester receives an HCI_Command_Status event in response.
8. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime</th>
<th>(Step 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0x14</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.33: Parameter values for each case variation
9. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The RSSI shall be set to a value in the range of -127 – 20 dBm. CTE_Type shall be set as specified in Table 4.34 (AoD Constant Tone Extension with 1 or 2 µs slots).

10. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT. Sampling_Enable shall be set to 0x01 (enabled). Max_Sampled_CTEs shall be set to 0x00 (Sample and report all available Constant Tone Extensions). Sync_Handle shall be set to the Sync_Handle from step 8. The Upper Tester receives an HCI_Command_Complete event from the IUT with Status set to 0x00 (Success) and Sync_Handle set to the Sync_Handle from step 8.

11. The Upper Tester receives two or more HCI_LE_Connectionless_IQ_Report events from the IUT. The Sync_Handle shall be set to a value in the range of 0x0000 - 0x0EFF. The Channel_Index shall be set to a value in the range of 0x00 - 0x24. The Packet_Status shall be set to 0x00 (CRC was correct). The RSSI shall be set to a valid value. The CTE_Type shall be set as specified in Table 4.34. The paEventCounter shall be set to the PeriodicEventCounter value from the SyncInfo field of the AUX_SYNC_IND. Sample_Count shall be set as specified in Table 4.34. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.

12. The Lower Tester ceases extended advertising but continues periodic advertising.

13. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report events and HCI_LE_Connectionless_IQ_Report events from the IUT as in steps 9 and 11.

14. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT to disable sampling.

15. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception.

16. The Upper Tester sends an HCI_LE_Set_EXTENDED_SCAN_Enable command to the IUT to disable scanning.

17. Repeat steps 1–16 for each round shown in Table 4.33.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Secondary Advertising PHY</th>
<th>CTE Type</th>
<th>LT Antenna Switching</th>
<th>Sample Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.2.3.24.1</strong> LL/DDI/SCN/BV-29-C [AoD Connectionless CTE Scanning – LE 1M PHY, 2 µs slots]</td>
<td></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02</td>
<td>0x13</td>
</tr>
<tr>
<td><strong>4.2.3.24.2</strong> LL/DDI/SCN/BV-48-C [AoD Connectionless CTE Scanning – LE 2M PHY, 2 µs slots]</td>
<td></td>
<td>0x02 (LE 2M PHY)</td>
<td>0x02</td>
<td>0x13</td>
</tr>
<tr>
<td><strong>4.2.3.24.3</strong> LL/DDI/SCN/BV-49-C [AoD Connectionless CTE Scanning – LE 1M PHY, 1 µs slots]</td>
<td></td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01</td>
<td>0x15</td>
</tr>
</tbody>
</table>
### AoD Connectionless CTE Scanning Test Cases

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Secondary Advertising PHY</th>
<th>CTE Type</th>
<th>LT Antenna Switching</th>
<th>Sample Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3.24.4 LL/DDI/SCN/BV-50-C [AoD Connectionless CTE Scanning – LE 2M PHY, 1 μs slots]</td>
<td>0x02 (LE 2M PHY)</td>
<td>0x01, 0x15, 0x01</td>
<td>1 μs slots</td>
<td>8 + (8 x CTETime - 12) / 2</td>
</tr>
</tbody>
</table>

*Table 4.34: AoD Connectionless CTE Scanning Test Cases*

- **Expected Outcome**

  **Pass Verdict**
  
  For all rounds described in the test procedure, the following condition shall occur:

  - The IUT generates two or more HCI_LE_Connectionless_IQ_Report events. 
    HCI_LE_Connectionless_IQ_Report events shall be sent for packets received with AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs.
  
  - The IUT maintains synchronization with the Lower Tester’s periodic advertisement PDUs and continues IQ sampling even after the Lower Tester ceases extended advertising.
  
  - The IUT terminates IQ sampling when the Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command.

  - In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 6, the Advertising_SID value in the event is the value sent by the Lower Tester in step 5.

  - The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.

### 4.2.3.25 AoA Connectionless CTE Scanning

- **Test Purpose**

  Tests that a scanner IUT detects and reports advertising events with the AoA Connectionless Constant Tone Extension included when advertiser is utilizing a public device address.

  The Lower Tester is advertising with an AoA Connectionless Constant Tone Extension, using a public device address. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- **Reference**

  [13] 2.5.2, 2.5.4

- **Initial Condition**

  Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, 
  LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

  State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

  The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.
**Test Procedure**

1. The Upper Tester sends an **HCI_LE_Read_Antenna_Information** command to the IUT and expects the IUT to return a `Max_Length_Switching_Pattern` between 0x02 and 0x4B. The Upper Tester stores the `Max_Length_Switching_Pattern` for future use.

---

**Figure 4.114: AoA Connectionless CTE Scanning**

For each round from 1 to 3

- **ADV_EXT_IND**
- **AUX_ADV_IND** (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)
- **ADV_EXT_IND**
- **AUX_ADV_IND** (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)
- **ADV_EXT_IND**
- **AUX_ADV_IND** (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)
- **AUX_CHAIN_IND** (One or more, CTE)
- **AUX_CHAIN_IND** (One or more, CTE)
- **AUX_CHAIN_IND** (One or more, CTE)
- **AUX_CHAIN_IND** (One or more, CTE)
For each round from 1–3 based on Table 4.35:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime (Step 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>0x14</td>
</tr>
</tbody>
</table>

Table 4.35: Parameter values for each case variation

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set as specified in Table 4.36, Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).

3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

4. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel set to LE 1M Phy (0x01) and AUX_ADV_IND PDUs on the secondary advertising channel as specified in Table 4.36. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address, a SyncInfo field referring to the AUXSYNC_IND PDUs and does not include the AdvData field. The Lower Tester continues advertising until directed to stop in the test procedure.

5. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel as specified in Table 4.36. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to the value specified in Table 4.35, RFU set to '0', and CTEType set to 0 (AoA Constant Tone Extension). The AUX_SYNC_IND PDUs shall not include the AdvData field. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in Table 4.35. The Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs.

6. The Lower Tester generates one or more AUX_CHAIN_IND PDUs on the secondary advertising channel as specified in Table 4.36. The AUX_CHAIN_IND PDUs shall not include the AdvData field. Each packet containing an AUX_CHAIN_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in Table 4.35.

7. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.

8. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements. Options shall be set to 0x00 (Don’t Use List). Advertising_SID shall be set to the Advertising_SID from step 7. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester’s address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 7. CTE_Type shall be set to 0x16 (accept only AoA Constant Tone Extension). The Upper Tester receives an HCI_Command_Status event in response.
9. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

10. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The RSSI shall be set to a value in the range of -127 – 20 dBm. CTE_Type shall be set to 0x00 (AoA Constant Tone Extension).

11. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT. Sampling_Enable shall be set to 0x01 (enabled). Max_Sampled_CTEs shall be set to 0x00 (Sample and report all available Constant Tone Extensions). Sync_Handle shall be set to the Sync_Handle from step 9. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, …, TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. Slot_Durations shall be set as specified in Table 4.36. The Upper Tester receives an HCI_Command_Complete event from the IUT with Status set to 0x00 (Success) and Sync_Handle set to the Sync_Handle from step 9.

12. The Upper Tester receives an HCI_LE_Connectionless_IQ_Report event from the IUT. The Sync_Handle shall be set to a value in the range of 0x0000 - 0x0EFF. The Channel_Index shall be set to a value in the range of 0x00 - 0x24. The Packet_Status shall be set to 0x00 (CRC was correct). The RSSI shall be set to a valid value. The RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array at step 11. The CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). Slot_Durations shall be set as specified in Table 4.36. The paEventCounter shall be set to the PeriodicEventCounter value from the SyncInfo field of the AUX_SYNC_IND. Sample_Count shall be set as specified in Table 4.36. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.

13. The Lower Tester ceases extended advertising but continues periodic advertising.

14. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report events and HCI_LE_Connectionless_IQ_Report events from the IUT as in steps 10 and 12.

15. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT to disable sampling.

16. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception.

17. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.

18. Repeat steps 2–17 for each round shown in Table 4.35.

- Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Secondary Advertising PHY</th>
<th>Slot Durations</th>
<th>Sample Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3.25.1 LL/DDI/SCN/BV-30-C [AoA Connectionless CTE Scanning – LE 1M PHY, 2 µs slots]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (2 µs slots)</td>
<td>0x02 (2 µs slots)</td>
</tr>
<tr>
<td>Test Case</td>
<td>Secondary Advertising PHY</td>
<td>Slot Durations</td>
<td>Sample Count</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Step 11</td>
<td>Step 12</td>
</tr>
<tr>
<td><strong>4.2.3.25.2</strong> LL/DDI/SCN/BV-51-C [AoA Connectionless CTE Scanning – LE 2M PHY, 2 µs slots]</td>
<td>0x02 (LE 2M PHY)</td>
<td>0x02 (2 µs slots)</td>
<td>0x02 (2 µs slots)</td>
</tr>
<tr>
<td><strong>4.2.3.25.3</strong> LL/DDI/SCN/BV-52-C [AoA Connectionless CTE Scanning – LE 1M PHY, 1 µs slots]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (1 µs slots)</td>
<td>0x01 (1 µs slots)</td>
</tr>
<tr>
<td><strong>4.2.3.25.4</strong> LL/DDI/SCN/BV-53-C [AoA Connectionless CTE Scanning – LE 2M PHY, 1 µs slots]</td>
<td>0x02 (LE 2M PHY)</td>
<td>0x01 (1 µs slots)</td>
<td>0x01 (1 µs slots)</td>
</tr>
</tbody>
</table>

*Table 4.36: AoA Connectionless CTE Advertising Test Cases*

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:

  - The IUT generates two or more HCI_LE_Connectionless_IQ_Report events.
    - HCI_LE_Connectionless_IQ_Report events shall be sent for packets received with AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs.
  
  - The IUT maintains synchronization with the Lower Tester’s periodic advertisement PDUs and continues IQ sampling even after the Lower Tester ceases extended advertising.
  
  - The IUT terminates IQ sampling when the Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command.
  
  - In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 7, the Advertising_SID value in the event is the value sent by the Lower Tester in step 4.
  
  - The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.

**4.2.3.26 AoD Connectionless CTE Scanning, Incorrect CRC**

- **Test Purpose**

  Tests that a scanner IUT detects and reports advertising events with the AoD Connectionless Constant Tone Extension included when receiving packets with incorrect CRCs.

  The Lower Tester is advertising with an AoD Connectionless Constant Tone Extension, using incorrect CRCs. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- **Reference**

  [13] 2.5.2, 2.5.4
Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

Test Procedure

Execute the Test Case as specified in Table 4.37 (AoD Connectionless CTE Scanning), except that:

In step 5, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC.

In step 9, the Upper Tester receives from the IUT one HCI_LE_Periodic_Advertising_Report event with the Data_Status set to 0x01 (Data incomplete, more data to come) and CTE_Type set to 0x02 (AoD Constant Tone Extension with 2 µs slots), and one HCI_LE_Periodic_Advertising_Report event with the Data_Status of 0x02 (Data incomplete, data truncated, no more to come) and CTE_Type set to 0x02 (AoD Constant Tone Extension with 2 µs slots) or 0xFF (No Constant Tone Extension).

In step 11, the Upper Tester receives from the IUT one or more HCI_LE_Connectionless_IQ_Report events with Packet_Status set to 0x01 or 0x02.

Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Test Case to Execute</th>
</tr>
</thead>
</table>

Table 4.37: AoD Connectionless CTE Scanning, Incorrect CRC Test Cases

Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connectionless_IQ_Report events with Packet_Status set to indicate the CRC was incorrect.
Inconclusive Verdict

The IUT only generates HCI_LE_Connectionless_IQ_Report events with Packet_Status set to indicate the CRC was correct.

4.2.3.27 AoA Connectionless CTE Scanning, Incorrect CRC

- **Test Purpose**

Tests that a scanner IUT detects and reports advertising events with the AoA Connectionless Constant Tone Extension included when receiving packets with incorrect CRCs.

The Lower Tester is advertising with an AoA Connectionless Constant Tone Extension, using incorrect CRCs. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- **Reference**

[13] 2.5.2, 2.5.4

- **Initial Condition**


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- **Test Procedure**

Execute the Test Case as specified in Table 4.38 (AoA Connectionless CTE Scanning), except that:

In step 6, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC.

In step 10, the Upper Tester receives from the IUT one HCI_LE_Periodic_Advertising_Report event with the Data_Status set to 0x01 (Data incomplete, more data to come) and CTE_Type set to 0x00 (AoA Constant Tone Extension), and one HCI_LE_Periodic_Advertising_Report event with the Data_Status of 0x02 (Data incomplete, data truncated, no more to come) and CTE_Type set to 0x00 (AoA Constant Tone Extension) or 0xFF (No Constant Tone Extension).

In step 12, the Upper Tester receives from the IUT one or more HCI_LE_Connectionless_IQ_Report events with Packet_Status set to 0x01 or 0x02.

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Test Case to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.2.3.27.1</strong> LL/DDI/SCN/BV-32-C [AoA Connectionless CTE Scanning, Incorrect CRC – LE 1M PHY, 2 µs slots]</td>
<td>LL/DDI/SCN/BV-32-C [AoA Connectionless CTE Scanning – LE 1M PHY, 2 µs slots]</td>
</tr>
<tr>
<td><strong>4.2.3.27.2</strong> LL/DDI/SCN/BV-57-C [AoA Connectionless CTE Scanning, Incorrect CRC – LE 2M PHY, 2 µs slots]</td>
<td>LL/DDI/SCN/BV-51-C [AoA Connectionless CTE Scanning – LE 2M PHY, 2 µs slots]</td>
</tr>
<tr>
<td>Test Case</td>
<td>Test Case to Execute</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.2.3.27.3 LL/DDI/SCN/BV-58-C [AoA Connectionless CTE Scanning, Incorrect CRC – LE 1M PHY, 1 µs slots]</td>
<td>LL/DDI/SCN/BV-52-C [AoA Connectionless CTE Scanning – LE 1M PHY, 1 µs slots]</td>
</tr>
<tr>
<td>4.2.3.27.4 LL/DDI/SCN/BV-59-C [AoA Connectionless CTE Scanning, Incorrect CRC – LE 2M PHY, 1 µs slots]</td>
<td>LL/DDI/SCN/BV-53-C [AoA Connectionless CTE Scanning – LE 2M PHY, 1 µs slots]</td>
</tr>
</tbody>
</table>

Table 4.38: AoA Connectionless CTE Scanning, Incorrect CRC Test Cases

- **Expected Outcome**

  **Pass Verdict**
  
The IUT generates HCI_LE_Connectionless_IO_Report events with Packet_Status set to indicate the CRC was incorrect.

  **Inconclusive Verdict**
  
The IUT only generates HCI_LE_Connectionless_IO_Report events with Packet_Status set to indicate the CRC was correct.

4.2.3.28 **Privacy - Extended Scanning, Active**

- **Test Purpose**

  Tests that a scanner IUT detects and reports a directed scannable advertising packet according to the scan filter policy set by the Upper Tester. The IUT also responds with a scan request and reports the scan response.

- **Reference**

  [10] 6.3, 4.4.2.8

- **Initial Condition**

  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT. Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

  State: Active Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner).

  The IUT is using a resolvable private address (0x02 or 0x03).

  The Lower Tester is also using a resolvable private address in the AdvA field of the advertising packets.

  The Lower Tester has previously distributed its IRK to the IUT.

- **Test Procedure**

  Execute the test procedure using the minimum advertising interval and continuous active scanning.
Figure 4.115: Network Privacy - Extended Scanning, Passive

1. The Upper Tester adds public address of the Lower Tester to the white list.
2. The Upper Tester adds peer device identity and local IRK information to resolving list.
3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x01 (Active Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x02 or 0x03 (Resolvable Private Address), and Scanning_Filter_Policy shall be set to 0x01 (Accept only advertising packets from devices where the advertiser’s address is in the White List).
4. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
5. The Lower Tester begins advertising using the ADV_EXT_IND PDU on the LE 1M PHY with AdvMode field set to 10b ("Scannable") and the AuxPtr field referencing the AUX_ADV_IND on the LE 1M PHY. The AUX_ADV_IND PDU shall include the AdvMode field set to 10b ("Scannable"), the AdvA field with a valid resolvable private address, and the TargetA field using a resolvable private address generated from a random IRK different from the one distributed to the IUT.

6. The IUT tries to resolve the address in the TargetA field by checking against its resolving list and does not find a match.

7. Repeat steps 5–6 for at least 20 advertising intervals

8. The Lower Tester begins advertising again with the TargetA field using a resolvable private address generated from the correct IRK.

9. Repeat step 8 for at least 20 advertising intervals or until step 11.

10. The Upper Tester receives one HCI_LE_Extended_Advertising_Report event from the IUT with an Event_Type where
    - bit 1 ("Scannable advertising") and bit 2 ("Directed Advertising") are set,
    - bit 0 ("Connectable advertising") and bit 4 ("Legacy advertising PDUs used") are cleared,
    - other bits may have any value.

11. The Lower Tester receives an AUX_SCAN_REQ on the appropriate advertising channel. The ScanA field shall be set to the IUT’s address and the AdvA address set to the Lower Tester’s address. The Upper Tester receives the Direct Address Type and Direct Address used to direct the advertisement at the IUT.

12. The Lower Tester responds with an AUX_SCAN_RSP packet to the IUT T_IFS after the end of the AUX_SCAN_REQ PDU with an AdvMode of 00b.

13. The Upper Tester receives one HCI_LE_Extended_Advertising_Report event from the IUT with an Event_Type where
    - bit 1 ("Scannable advertising"), bit 2 ("Directed Advertising") and bit 3 ("Scan response") are set,
    - bit 0 ("Connectable advertising"), bit 4 ("Legacy advertising PDUs used") and bits 5-6 ("Data Status") are cleared,
    - other bits may have any value.

14. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

- **Expected Outcome**

  **Pass Verdict**

  The IUT receives scannable directed advertising with a resolvable private address in the AdvA field and an invalid or valid resolvable private address in the TargetA field from the Lower Tester.

  For advertisements where TargetA address does not match the resolving list, the IUT does NOT generate a scan request and an HCI_LE_Extended_Advertising_Report event.

  For advertisements where TargetA address matches the resolving list, the IUT generates a scan request and an HCI_LE_Extended_Advertising_Report event.

  In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 10, the Advertising_SID value in the event is the value sent by the Lower Tester in step 5.

  In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 13, the Advertising_SID value in the event is the value as specified in Table 4.39.
<table>
<thead>
<tr>
<th>Test Case</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.2.3.28.1</strong> LL/DDI/SCN/BV-33-C Privacy - Extended Scanning, Active, Core 5.0</td>
<td>0xFF or the Advertising SID from the AUX_ADV_IND PDU.</td>
</tr>
<tr>
<td><strong>4.2.3.28.2</strong> LL/DDI/SCN/BV-63-C Privacy - Extended Scanning, Active, Core 5.1</td>
<td>Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU.</td>
</tr>
</tbody>
</table>

Table 4.39: Privacy - Extended Scanning, Active test cases

**4.2.3.29** LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]

- **Test Purpose**
  Tests that a scanner IUT can synchronize to periodic advertising events using both periodic sync establishment filter policy settings, and ignoring the scanner filter policy.

- **Reference**
  [10] 4.3.5, 4.4.3.4

- **Initial Condition**
  State: Idle.
  The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.
• Test Procedure

Lower Tester

IUT

Upper Tester

- HCI_LE_Add_Device_To_White_List (Random Address)
  - HCI_Command_Complete (Status: 0x00)
  - HCI_LE_Set_Extended_Scan_Parameters (Extended Scanning)
    - HCI_Command_Complete (Status: 0x00)
    - HCI_LE_Set_Extended_Scan_Enable (Enable)
      - HCI_Command_Complete (Status: 0x00)

- HCI_LE_Periodic_Advertising_Create_Sync (Filter Policy 0x00)
  - HCI_Command_Status (Status: 0x00)
  - HCI_LE_Periodic_Advertising_Sync_Established (Status: 0x00)
  - HCI_LE_Periodic_Advertising_Report (One or more)
    - HCI_Command_Complete (Status: 0x00)

- Upper Tester expects no report

- AUX_ADV_IND
  - (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)

- AUX_SYNC_IND
  - (Secondary channel, AdvMode: 00b, AdvA, SyncInfo)

Continued in Part B...
Continued from Part A...

AUX_SYNC_IND

HCI_LE_Clear_Periodic_Advertiser_List

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Add_Device_To_Periodic_Advertiser_List (Lower Tester’s Address)

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Periodic_Advertising_Create_Sync (Filter Policy: 0x01)

HCI_Command_Start
(Status: 0x00)

HCI_LE_Periodic_Advertising_Sync_Established
(Status: 0x00)

HCI_LE_Periodic_Advertising_Report
(One or more)

HCI_LE_Periodic_Advertising_Terminate_Sync

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Clear_Periodic_Advertiser_List
(Status: 0x00)

HIC_Lcommand_Start
(Status: 0x00)

HCI_LE_Periodic_Advertising_Create_Sync (Filter Policy: 0x01)
(Status: 0x00)

AUX_SYNC_IND

HCI_LE_Add_Device_To_Periodic_Advertiser_List (Different from Lower Tester’s Address)

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Periodic_Advertising_Create_Sync

HCI_Command_Start
(Status: 0x00)

Upper Tester expects no Sync Establishment

HCI_LE_Periodic_Advertising_Create_Sync_Cancel

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Clear_Periodic_Advertiser_List

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Add_Device_To_Periodic_Advertiser_List (Lower Tester’s Address, Different Adv SID)

HCI_Command_Complete
(Status: 0x00)

HCI_LE_Periodic_Advertising_Create_Sync

HCI_Command_Start
(Status: 0x00)

Upper Tester expects no Sync Establishment

HCI_LE_Periodic_Advertising_Create_Sync_Cancel

HCI_Command_Complete
(Status: 0x00)

Figure 4.116: LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]
1. The Upper Tester sends an HCI_LE_Add_Device_To_White_List command to the IUT, containing a randomly generated device address that is different than the Lower Tester's address and receives an HCI_Command_Complete event in response.

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x01 (use White List). The Upper Tester receives an HCI_Command_Complete event in response.

3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero. The Upper Tester receives an HCI_Command_Complete event in response.

4. The Lower Tester begins non-connectable, non-scannable, undirected advertising using ADV_EXT_IND and AUX_ADV_IND PDUs, on the LE 1M PHY. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester's address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

5. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in step 3. The advertising data is chosen to fit in a single AUX_SYNC_IND PDU. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

6. The Upper Tester sends an HCI_LE_Periocic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Filter_Policy shall be set to 0x00 (Don't Use List). Advertiser_SID shall be set to the value used by the Lower Tester in the periodic advertisements. Advertiser_Address_Type and Advertiser_Address shall be set to match the Lower Tester's address. Skip shall be set to 0x0000. Sync_Timeout shall be set to three times the periodic advertising interval. Unused shall be set to 0x00. The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The Upper Tester receives the data included in the advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to 0xFF.

9. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

10. Upper Tester sends an HCI_LE_Clear_Periodic_Advertiser_List command, followed by an HCI_LE_Add_Device_To_Periodic_Advertiser_List command containing the Lower Tester's address and the Advertising SID used in the periodic advertisements. The Upper Tester receives an HCI_Command_Complete event in response to each command.

11. Repeat steps 6–9, but in step 6 the Filter_Policy is set to 0x01 (Use Advertiser List).

12. Repeat step 10 using a different advertiser address.

13. Repeat step 6 with the Filter_Policy set to 0x01. The Upper Tester expects the IUT not to synchronize with the periodic advertisements of the Lower Tester.

14. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command and receives an HCI_Command_Complete event in response (indicating success).

15. Repeat step 10 using the correct advertiser address but a different Advertising SID.

16. Repeat step 6 with the Filter_Policy set to 0x01. The Upper Tester expects the IUT not to synchronize with the periodic advertisements of the Lower Tester.
17. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command and receives an HCI_Command_Complete event in response (indicating success).

• Expected Outcome

Pass Verdict
After the first and second times that step 6 is executed, the IUT is able to synchronize to the periodic advertisements. After the third and fourth times, the IUT does not synchronize to the periodic advertisements.

Fail Verdict
The IUT generates extended advertising reports at any time throughout the test.

4.2.3.30 LL/DDI/SCN/BV-35-C [Connectionless CTE Scanning, Filter Wrong CTE Types on Synchronization]

• Test Purpose
Tests that a scanner IUT applies the filtering policy while attempting synchronization with the periodic advertising, when the Lower Tester uses a wrong Constant Tone Extension type of the periodic advertisements.

• Reference
[13] 2.5.2, 2.5.4

• Initial Condition
State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).
1. The Upper Tester sends an HCI_LE_Set_EXTENDED_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_EXTENDED_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address.
and a SyncInfo field referring to the AUX_SNYC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to any valid value, RFU set to ‘0’, and CTEType set to a valid value for a CTE type that the IUT supports. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements. Options shall be set to 0x00 (Don’t Use List). Advertising_SID shall be set to the Advertising_SID from step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester’s address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 5. CTE_Type shall be set to a value that specifies not to synchronize to packets with a CTE type from step 4. The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x1A (Unsupported Remote Feature / Unsupported LMP Feature) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester does not receive any HCI_LE_Periodic_Advertising_Report events from the IUT.

9. The Upper Tester sends an HCI_LE_Add_Device_To_Periodic_Advertiser_List command. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to a valid address other than the Lower Tester’s address. Advertising_SID shall be set to a valid value other than the Advertising_SID from step 5. The Upper Tester receives an HCI_Command_Status event in response.

10. Repeat step 6 with Options set to 0x01 (Use List).

11. The Upper Tester does not receive any HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) or HCI_LE_Periodic_Advertising_Report events from the IUT.

12. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command to the IUT to cancel the pending HCI_LE_Periodic_Advertising_Create_Sync command and receives an HCI_Command_Status event in response.

13. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:
  - The IUT does not synchronize with the Lower Tester’s periodic advertisement PDUs.
  - When filtering policy is set to 0x01, the IUT continues synchronization until the procedure is canceled by the Upper Tester.
  - When filtering policy is set to 0x00, the IUT stops the synchronization procedure.
  - In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value sent by the Lower Tester in step 3.
4.2.3.31 LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]

- Test Purpose
  Tests that a scanner IUT maintains synchronization with the periodic advertising, when the Lower Tester changes the Constant Tone Extension type or does not transmit any Constant Tone Extension of in the periodic advertisements.

- Reference
  [13] 2.5.2, 2.5.4

- Initial Condition
  State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

- Test Procedure

  ![Diagram](image-url)

  Figure 4.118: LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]
1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to any valid value, RFU set to ‘0’, and CTEType set to a valid value for a CTE type that the IUT supports. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_AdvertisingInterval.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements. Options shall be set to 0x00 (Don’t Use List). Advertising_SID shall be set to the Advertising_SID from step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester’s address. Sync_Timeout shall be set to 3 x Periodic_AdvertisingInterval from step 5. CTE_Type shall be set to a value that specifies to synchronize to packets with a CTE type from step 4. The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to a valid value for a CTE type that the IUT supports.

9. The Lower Tester ceases extended advertising but continues periodic advertising.

10. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report from the IUT as in step 8.

11. The Lower Tester generates AUX_SYNC_IND PDUs as in step 4, but with CTE_Type set to a valid value for a CTE type other than the value from step 4.

12. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to the same value from step 11.

13. The Lower Tester generates AUX_SYNC_IND PDUs as in step 4, but without a CTEInfo field and a Constant Tone Extension.

14. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to 0xFF (No Constant Tone Extension).

15. Repeat step 4. The Upper Tester continues to receive HCI_LE_Periodic_Advertising_Report from the IUT as in step 8.

16. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.
• Expected Outcome
  
  **Pass Verdict**
  
  - The IUT maintains synchronization with the Lower Tester’s periodic advertisement PDUs even after the Lower Tester ceases extended advertising.
  
  - The IUT maintains synchronization with the Lower Tester’s periodic advertisement PDUs after the Lower Tester or changes the Constant Tone Extension Type for the AUX_SYNC_IND PDUs or ceases transmitting a Constant Tone Extension with the AUX_SYNC_IND packet.
  
  - In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value sent by the Lower Tester in step 3.

**4.2.3.32 LL/DDI/SCN/BV-37-C [First AUX_SYNC_IND Never Received]**

• Test Purpose
  
  Tests that synchronization will fail on the IUT when an AUX_SYNC_IND PDU is not received within 6 periodic advertising events after the first advertising event received by the IUT.

• Reference
  
  [10] 4.4.3.4

• Initial Condition
  

  State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

  The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

• Test Procedure
  
  Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.
1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT. Options shall be set to 0x00 (Don't Use List). Advertising_SID, Advertiser Address_Type and Advertiser_Address shall be set to those of the Lower Tester. Skip shall be set to a valid Skip value. Sync_Timeout shall be set to (Skip + 3) x Periodic_Advertising_Interval. Sync_CTE_Type shall be set to 0x00. The Upper Tester receives an HCI_Command_Status event in response.

3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

4. The Lower Tester begins advertising using ADV_EXT_IND and AUX_ADV_IND PDUs using the LE 1M PHY with a Periodic Advertising Interval of 150 ms and Advertising Interval of 50 ms. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester address and a SyncInfo field referring to a valid sequence of AUX_SYNC_IND PDUs.
PDUs. The Lower Tester does not transmit the AUX_SYNC_IND PDUs initially. The Lower Tester continues advertising until directed to stop in the test procedure.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval, when it first receives and AUX_ADV_IND from the Lower Tester, also indicating when the synchronization attempt begins.

6. The Lower Tester sends AUX_SYNC_IND PDUs starting with the 6th periodic advertising event after the Upper Tester receives the HCI_LE_Extended_Advertising_Report event.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event with Status set to 0x3E (Connection Failed To Be Established / Synchronization Timeout).

- Expected Outcome

**Pass Verdict**
- The IUT sends HCI_LE_Periodic_Advertising_Sync_Established event ceasing attempt to synchronize after sending the HCI_LE_Extended_Advertising_Report event in step 5 and reports to the Upper Tester.
- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 4, the Advertising_SID value in the event is the value sent by the Lower Tester in step 3.

**Fail Verdict**
The IUT successfully synchronizes to the periodic advertising sent in step 6.

### 4.2.3.33 Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled

- **Test Purpose**
  Tests that a scanner IUT can locate and receive periodic advertising events when reporting is initially disabled. The Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising, and the Upper Tester expects the IUT to report sync establishment without generating periodic advertising reports until reporting is enabled and stops generating reports after reporting is disabled.

- **Reference**
  [13] 4.4.2.13.1, 4.4.3.4

- **Initial Condition**
  State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

- **Test Procedure**
  Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.
Figure 4.120: Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT with Scanning_PHYs set as specified in Table 4.40, Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type is set to 0x00 (Public Device Address), and Scanning_Filter_Policy is set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period are all set to zero.

3. The Lower Tester begins advertising using ADV_EXT_IND and AUX_ADV_IND PDUs using the PHY as specified in Table 4.40. The ADV_EXT_IND PDUs includes an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs includes the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SNYSYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Lower Tester generates ADV_SYNC_IND PDUs on the secondary advertising channel as specified in Table 4.40 using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in step 3. The Lower Tester continues periodic advertising until directed to stop in the test procedure.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing a non-zero Periodic_Advertising_Interval.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements. Options is set to 0x02 (Don’t Use List, Reporting Initially Disabled)”. Advertising_SID is set to the Advertising_SID from step 5. Advertiser_Address_Type is set to 0x00 (Public Device Address). Advertiser_Address is set to
the Lower Tester’s address. Skip is set to 0. Sync_Timeout is set to 3 x Periodic_Advertising_Interval from step 5. CTE_Type is set to 0x00 (don’t care). The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester receives no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

9. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x01 (Reporting enabled) and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

10. The Upper Tester receives one or more HCI_LE_Periodic_Advertising_Report events from the IUT.

11. Repeat step 10 until at least 30 HCI_LE_Periodic_Advertising_Report events have been received.

12. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x00 (Reporting disabled) and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

13. The Upper Tester receives no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

14. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

15. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

16. The Lower Tester ceases periodic advertising.

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Scanning PHY</td>
</tr>
<tr>
<td>4.2.3.33.1 LL/DDI/SCN/BV-38-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.2.3.33.2 LL/DDI/SCN/BV-60-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled – LE 2M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.2.3.33.3 LL/DDI/SCN/BV-61-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled – LE Coded PHY]</td>
<td>0x04 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

*Table 4.40: Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled Test Cases*
• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following conditions occur:

- The IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements using the Channel Selection Algorithm #2.

- The IUT does not generate HCI_LE_Periodic_Advertising_Report events when reporting is disabled.

- The IUT generates HCI_LE_Periodic_Advertising_Report events when reporting is enabled.

- The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value sent by the Lower Tester in step 3.

4.2.3.34 LL/DDI/SCN/BI-01-C [Active Scanning Invalid CRC]

• Test Purpose

Tests that a scanner IUT ignores advertising indication packets with invalid CRCs.

The Lower Tester sends advertising packets with invalid CRCs and checks that the IUT does not send a scan request. The Lower Tester sends also scan response packets with invalid CRCs and observes the HCI events from the IUT.

• Reference

[3] 3.1, 4.4.3.2

• Initial Condition


State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))

• Test Procedure

Execute the test procedure using the minimum advertising interval and the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.
1. **Upper Tester** enables active scanning in the **IUT**.
2. Configure **Lower Tester** advertising channel map using the first supported advertising channel.
3. Configure **Lower Tester** as advertiser using a common device address with invalid checksum.
4. **Lower Tester** sends ADV_IND packets with an invalid checksum, each advertising event, using the selected advertising interval. Repeat until the time exceeds a number of scan intervals (20).
5. **Lower Tester** receives no SCAN_REQ packets T_IFS after any of the ADV_IND packets.
6. **Upper Tester** receives no **HCI_LE_Advertising_Report Event** reporting the advertising packets sent by the **Lower Tester** in step 4.
7. Configure **Lower Tester** as advertiser using a common device address (different than device address used in step2) with correct checksum.
8. **Lower Tester** sends ADV_IND packets with correct checksum, each advertising event, using the selected advertising interval. Repeat until the time exceeds a number of scan intervals (20) or step 9 executes.
9. **Lower Tester** receives an SCAN_REQ packet T_IFS after any of the ADV_IND packets.
10. **Lower Tester** sends an SCAN_RSP packet with an invalid checksum to the **IUT** T_IFS after the SCAN_REQ packet.
11. Interleave with step 8, **Upper Tester** receives an **HCI_LE_Advertising_Report Event** containing the advertising packets information.

---

**Figure 4.121: LL/DDI/SCN/BI-01-C [Active Scanning: Invalid CRC]**
13. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.
14. Configure Lower Tester advertising channel map using the second supported advertising channel.
16. Configure Lower Tester advertising channel map using the third supported advertising channel.

- **Expected Outcome**
  
  **Pass Verdict**
  
  The test procedure completes using each supported advertising channel with the timing combination, with the IUT ignoring the invalid advertising channel packets.

- **Notes**

  In order to avoid interference, the test procedure uses the filtering policy to blacklist all unknown devices (0x01) if device filtering is supported. The features are assumed to be independent therefore the test can be executed without the filtering in conditions with LE traffic.

4.2.3.35  **LL/DDI/SCN/BI-02-C [Passive Scanning Invalid CRC]**

- **Test Purpose**

  Tests that a scanner IUT ignores data in advertising packets with invalid checksums.

  The Lower Tester sends advertising packets on several channels, expecting the IUT not to report the data with invalid checksums.

- **Reference**

  [3] 3.1, 4.4.3.1

- **Initial Condition**

  Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
  LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

  State: Passive Scanning (selected scan interval, selected scan window AND Specific White Listed (Lower Tester address, one public type address, policy for scanner, blacklist all unknown devices) OR All White Listed (policy for scanner)).

- **Test Procedure**

  Execute the test procedure with a minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).
1. Upper Tester enables passive scanning in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel.
3. Configure Lower Tester as advertiser using a common device address with invalid checksum.
4. Lower Tester sends ADV_NONCONN_IND packets with an invalid checksum, each advertising event. Repeat for a time that exceeds a number of scan intervals (20).
5. Upper Tester receives no HCI_LE_Advertising_Report events containing the advertising packets information used in step 4.
6. Configure Lower Tester advertising channel map using the second supported advertising channel.
7. Configure Lower Tester as advertiser using a common device address with invalid checksum.
8. Repeat steps 4–6.
9. Configure Lower Tester advertising channel map using the third supported advertising channel.
10. Configure Lower Tester as advertiser using a common device address with invalid checksum.
11. Repeat steps 4–6.
12. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and receive an HCI_Command_Complete event in response.
• **Expected Outcome**

  **Pass Verdict**

  The test procedure completes using each supported advertising channel separately.

• **Notes**

  In order to avoid interference, the test procedure uses the filtering policy to black list all unknown devices (0x01) if device filtering is supported. The features are assumed to be independent and the test can be executed without the filtering in conditions with LE traffic.

### 4.2.3.36 LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Wrong AdvA in Response]

• **Test Purpose**

  Verify that the IUT when doing active scanning ignores a SCAN_RSP that does not contain the same AdvA used in the SCAN_REQ.

• **Reference**

  [3] 4.4.3.2

• **Initial Condition**

  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

  State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices))
• Test Procedure

1. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.
2. The Upper Tester adds peer device identity and local IRK information to resolving list. Using the HCI_LE_Add_Device_To_Resolving_List and HCI_LE_Set_Address_Resolution_Enable commands.
3. The Upper Tester adds the address of the Lower Tester to the white list.
4. The Upper Tester enables active scanning with filtering policy set to ‘Accept only advertising packets from devices where the advertiser’s address is in the White List (0x01)’ in the IUT.
5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
6. The Lower Tester receives a SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. The ScanA field in the SCAN_REQ packet shall use a resolvable private address.
7. The Lower Tester sends a SCAN_RSP packet T_IFS after the SCAN_REQ packet. The AdvA field in the SCAN_RSP packet shall use a resolvable private address that does not match the one used in the SCAN_REQ packet. The AdvA field in the SCAN_RSP is generated using an IRK different from the IUT’s IRK. The Upper Tester does not expect the IUT to send an HCI_LE_Advertising_Report event containing the information in the SCAN_RSP packet.

Figure 4.123: LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Local IRK, Peer IRK]
8. The Lower Tester optionally receives an additional SCAN_REQ packet from the IUT.
9. Interleave with step 5: Upper Tester receives an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
10. The Upper Tester disables scanning.

• Expected Outcome

**Pass Verdict**

The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester. The IUT successfully resolves the address.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The Lower Tester successfully resolves the address.

The IUT receives a SCAN_RSP packet with an AdvA field that does not match the address from the SCAN_REQ packet. The IUT is expected not to send an HCI_LE_Advertising_Report event to the Upper Tester.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.37 LL/DDI/SCN/BV-67-C [Periodic Advertising Reception, Channel Map Update]

• Test Purpose

Tests that a scanner IUT can locate and receive periodic advertising events, report the results from the Controller, and follow a Channel Map Update Indication provided by the Lower Tester.

• Reference

[10] 4.4.2.13.1, 4.4.3.4
[18] 1.20

• Initial Condition


State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

All primary and secondary channels are enabled in the channel map.

• Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.
1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter is set to the LE 1M PHY. Scan_Type[0] is set to 0x00 (Passive Scanning), Scan_Interval[0] is set to 0x0010, and Scan_Window[0] is set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.

3. The Lower Tester begins advertising using ADV_EXT_IND and AUX_ADV_IND PDUs. All advertising is transmitted on the LE 1M PHY. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in step 3. The AUX_SYNC_IND PDUs shall contain an AdvData field of length 31 octets consisting of random octets from 1 to 255.

5. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall
be set to the Lower Tester’s address. Skip shall be set to 0x0000. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 5. Sync_CTE_Type shall be set to 0x00. The Upper Tester receives an HCI_Command_Status event in response.

7. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.

8. The Upper Tester receives an HCI_LE_Periodic_Advertising_Report event from the IUT.

9. The Lower Tester ceases extended advertising but continues periodic advertising.

10. The Lower Tester begins sending AUX_SYNC_IND PDUs with an extended header that contains a Channel Map Update Indication with a channelMapNEW that only uses even channels, for a minimum of 6 events prior to the instant.

11. At the instant, the Lower Tester begins using channelMapNEW and continues periodic advertising.

12. The Upper Tester continues to receive HCI_LE_Periodic_AdvertiSing_Report events from the IUT as in step 8.

- Expected Outcome

  Pass Verdict

  The IUT maintains synchronization with the Lower Tester periodic advertisement PDUs after the Channel Map Update Indication instant.

4.3 CON

Tests that the IUT behaves according to the connection setup and connection procedures.

4.3.1 Common PDU Contents

The packet descriptions for advertising and data channel packets sent and accepted by the Lower Tester are displayed below.

The addresses used in tests may vary for the Lower Tester,

The device address for the IUT is expected to match the IXIT value entered or set in the test preambles.

CONNECT_IND PDU:

<table>
<thead>
<tr>
<th>Header</th>
<th>Length</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
<td>LSO MSO Isb msb</td>
</tr>
<tr>
<td>Type ‘1010’</td>
<td>RFU ‘0’</td>
<td>ChSel ‘0’</td>
</tr>
<tr>
<td></td>
<td>ChSel ‘0’</td>
<td>TxAdd ‘0’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RxAdd ‘0’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Len ‘01000100’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add (16 octets)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LLData (18 octets)</td>
</tr>
</tbody>
</table>

CONNECT_IND Payload Add Field:

<table>
<thead>
<tr>
<th>LSO MSO Isb msb</th>
<th>LSO MSO Isb msb</th>
<th>LSO MSO Isb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitA (6 octets)</td>
<td>AdvA (6 octets)</td>
<td>AA (4 octets)</td>
</tr>
</tbody>
</table>
CONNECT_IND Payload LLData Field:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRCInit</td>
<td>WinSize</td>
<td>WinOffset</td>
<td>Interval</td>
<td>Latency</td>
<td>Timeout</td>
<td>ChM</td>
<td>Hop</td>
<td>SCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3 octets)</td>
<td>‘11100000’</td>
<td>‘11100000’</td>
<td>‘00010000’</td>
<td>‘00000000’</td>
<td>‘01111000’</td>
<td>‘10000000’</td>
<td>‘10100’</td>
<td>‘000’</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The connection supervision timer value of 300 ms is based on the close to zero probability that an empty data packet would be received incorrectly around 10 (or more) times, with a slave device listening for every 3rd master transmission, given the BER of 0.1%.

CONNECTION_UPDATE_IND CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
<th>LSO MSO</th>
<th>Isb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>WinSize</td>
<td>WinOffset</td>
<td>Interval</td>
<td>Latency</td>
<td>Timeout</td>
<td>Instant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1 octet)</td>
<td>(2 octets)</td>
<td>(2 octets)</td>
<td>(2 octets)</td>
<td>(2 octets)</td>
<td>(2 octets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FEATURE_REQ CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>Isb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>FeatureSet</td>
<td>(8 octets)</td>
</tr>
</tbody>
</table>

FEATURE_RSP CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>Isb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>FeatureSet</td>
<td>(8 octets)</td>
</tr>
</tbody>
</table>
4.3.2 ADV
Tests that the IUT behaves according to the connection setup procedures as an advertiser.

4.3.2.1 Common PDU Contents
CONNECT_IND Payload LL Data Field:

<table>
<thead>
<tr>
<th>LSO MSO lsb msb</th>
<th>lsb msb</th>
<th>lsb msb</th>
<th>LSO MSO lsb msb</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRCInit '10001001 0001001 10011000'</td>
<td>Interval '00010000'</td>
<td>Hop '10100'</td>
<td>SCA '000'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ChM '10000000 00000000 00000000 00000000 0000'</td>
</tr>
</tbody>
</table>

4.3.2.2 Common Variables
Master and slave roles in the connection handling tests use variables indicating the state of acknowledgements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial Value</th>
<th>Constraints</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>current SN</td>
<td>Assigned from the SN in a packet sent.</td>
<td>The range is modulo 2.</td>
<td>The sequence number matching a packet sent.</td>
</tr>
<tr>
<td>current NESN</td>
<td>Assigned from the NESN in a packet sent.</td>
<td>The range is modulo 2.</td>
<td>The next expected sequence number to match a packet to be received.</td>
</tr>
<tr>
<td>next SN</td>
<td>0</td>
<td>The range is modulo 2, value current SN + 1.</td>
<td>The next sequence number to be sent.</td>
</tr>
<tr>
<td>next NESN</td>
<td>0</td>
<td>The range is modulo 2, value current NESN + 1.</td>
<td>The next expected sequence number to be sent.</td>
</tr>
</tbody>
</table>

The current values reflect a packet sent to the IUT. The values assigned to a packet are dependent on the test and the response packets from the IUT.

4.3.2.3 LL/CON/ADV/BV-01-C [Accepting Connections]

- Test Purpose
  Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role.

  The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role, observing the packets and timing from the IUT.
• Reference

[3] 4.5.5

• Initial Condition

Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map

State: Undirected Advertising (selected Adv Interval_Min, selected Adv Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices))

• Test Procedure

Execute the test procedure using the common data channel selection parameters and connection parameters to setup the connection.

![Test Procedure Diagram]

Figure 4.125: LL/CON/ADV/BV-01-C [Accepting Connections]

1. Configure Lower Tester to initiate a connection.
2. Upper Tester enables advertising in the IUT using the first supported advertising channel.
3. Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only, using the selected advertising interval.
4. Repeat 20 times or until the IUT stops advertising: Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet TIFS after the end of the advertising packet.
5. Lower Tester sends a correctly formatted LL Data Channel PDU starting the first event at connection interval after the connection request using the common data channel selection parameters.
6. Lower Tester receives a correctly formatted LL Data Channel PDU from the IUT TIFS after the PDU sent on the same data channel. Lower Tester continues sending correctly formatted LL Data
Channel PDUs until receiving a number of responses (100) to conclude the timing accuracy or up to 900 events transmitted.

7. Interleave with step 6: Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 4.

8. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 4).

9. Configure Lower Tester to initiate a connection.

10. Upper Tester enables advertising in the IUT using the second supported advertising channel.


12. Configure Lower Tester to initiate a connection.

13. Upper Tester enables advertising in the IUT using the third supported advertising channel.


• Expected Outcome

**Pass Verdict**

The test procedure completes with the IUT stopping advertising by a connection requested from all advertising channels, moving to the data channel and responding in the connection events.

The number of timing measurements for the reply packets from the IUT is at least 100 and the timing deviations detected for packets in active mode are within the 2 µs range around T_IFS.

The IUT reports the connection setup with an HCI event.

The IUT maintains the connection using the Channel Selection Algorithm #1.

• Notes

The timing observation criterion takes into account active mode requirements. For the slave device the low power mode requirement is not tested here. The active mode timing in packet exchange is listed in the pass verdict to be within the 2 µs range around T_IFS.

The accuracy required is 1 µs, from the expressions 150 µs and 2 µs. Drift does not affect measurements in exchanges of single data packets. Measurement accuracy is at least 0.1 µs and rounding is done to microseconds.

In order to avoid interference, the test procedure uses the filtering policy to Allow Scan Request from White List, Allow Connect Request from White List if device filtering is supported. The features are assumed to be independent and the test can be executed without the filtering in conditions with low LE traffic.

**4.3.2.4 LL/CON/ADV/BV-02-C [Accepting Connections Timeout]**

• Test Purpose

Tests that an advertiser IUT receives a connection request, stops advertising after the reception and after receiving no master transmission before connection supervision timer expiration, reports the connection setup failed.

The Lower Tester acts in the initiating state, sending the connection request to the IUT then does not begin to maintain the connection and observes the IUT reports in HCI events.

• Reference

[3] 4.5.5
• Initial Condition
Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map.
State: Undirected Advertising (selected AdvInterval_Min, selected AdvInterval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices))

• Test Procedure
Execute the test procedure using the common data channel selection parameters and the common connection parameters to setup the connection.

1. Configure Lower Tester to initiate a connection.
2. Upper Tester enables advertising in the IUT using a selected supported advertising channel (defined as an IXIT).
3. Repeat 20 times or until the IUT stops advertising: Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
4. Lower Tester stops sending packets after CONNECT_IND packet and waits until connection supervision timeout timer expires.
5. Interleave with step 4: Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.
6. Upper Tester receives an HCI_Disconnection_Complete event from the IUT with the reason parameter indicating ‘connection failed to be established’, with the connection handle parameter matching to step 5.

Figure 4.126: LL/CON/ADV/BV-02-C [Accepting Connections Timeout]
• Expected Outcome
  **Pass Verdict**
  The test procedure completes with the IUT stopping advertising and reporting the failure to establish connection.

• Notes
  The timeout accuracy is not observed here, but the parameter value may be used in test implementation.

4.3.2.5  LL/CON/ADV/BV-03-C [Master Missing Slave Packets]

• Test Purpose
  Tests that an advertiser IUT after accepting a connection request, starts to maintain a connection in the slave role not taking slave latency into use before receiving an acknowledgement from the master.
  The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role. The Lower Tester first sends a number of negative acknowledgements to the IUT, then changes to observe the acknowledgement scheme.

• Reference
  [3] 4.5.5, 4.5.9

• Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Undirected Advertising (selected Adv_Inteval_Min, selected Adv_INTERVAL_MAX, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (one white listed device address, one public type address, policy for advertiser, black list all unknown devices))

• Test Procedure
  Execute the test procedure using the common data channel selection parameters and the common connection parameters to setup the connection, use a latency parameter value of 5.
  If device filtering is supported, use the filtering policy to white list all unknown devices (apply the filtering policy of allowing scan request, connect request from any).
1. Configure Lower Tester to initiate a connection.
2. Upper Tester enables advertising in the IUT using a selected supported advertising channel (defined as an IXIT).
3. Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
4. Lower Tester sends a correctly formatted LL Data Channel PDU, not acknowledging the slave packets, starting the first event using the common connection interval timing after the connection request.
5. Lower Tester receives a correctly formatted LL Data Channel PDU from the IUT, T_IFS after the PDU sent on the same data channel. Lower Tester continues sending correctly formatted LL Data Channel PDUs with the default timing.
6. Repeat steps 4–5 for a number of events (30).
7. Lower Tester sends a correctly formatted LL Data Channel PDU acknowledging the slave packets, using the common connection interval timing and the common data channel selection parameters.
8. Lower Tester receives a correctly formatted LL Data Channel PDU from the IUT, T_IFS after the PDU sent on the same data channel, for events between the common slave latency parameter.

9. Repeat steps 7–8 for a number of events (30).

10. Interleave with steps 4–5: Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.

11. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from 10).

• Expected Outcome

   Pass Verdict

   The test procedure completes with the IUT stopping advertising, moving to the data channel and responding in the connection events.

   The IUT responds in at least 9 out of 30 connection events until the first master packet acknowledging the slave transmissions,

• Notes

   The figure for the percentage of responses required is based on the correctly formatted LL Data Channel PDUs transmitted to and from the slave (around 85% probability of correct receptions).

4.3.2.6 LL/CON/ADV/BV-04-C [Directed Advertising Connection]

• Test Purpose

   Tests that an advertiser IUT upon receiving a connection request to the directed advertising indications, stops advertising after the reception and starts to maintain a connection in the slave role.

   The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role.

• Reference

   [3] 4.4.2.4

• Initial Condition

   Parameters: LL_advertiser_Adv_Channel_Map

   State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)
Test Procedure

1. Configure Lower Tester to initiate a connection using a mismatching init address.
2. Upper Tester enables directed advertising in the IUT using all supported advertising channels.
3. Lower Tester receives an ADV_DIRECT_IND packet from the IUT and responds with a CONNECT_IND packet (with the mismatching init address) T_IFS after the end of the advertising packet.
4. Lower Tester receives ADV_DIRECT_IND packets after the connection request. Repeat up to a period of time (1.28 s) or until the IUT stops advertising.
5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT with status parameter set to 'directed advertising timeout'.
6. Configure Lower Tester to initiate a connection.
7. Upper Tester enables directed advertising in the IUT using all supported advertising channels.
8. Lower Tester receives a ADV_DIRECT_IND packet from the IUT and responds with a CONNECT_IND packet (with a matching init address) T_IFS after the end of the advertising packet.
9. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 8.
10. Lower Tester sends a correctly formatted LL Data Channel PDU starting the first event at connection interval after the connection request using the common data channel selection parameters.
11. Lower Tester receives a correctly formatted LL Data Channel PDU from the IUT, T_IFS after the correctly formatted LL Data Channel PDU sent on the same data channel and continues sending correctly formatted LL Data Channel PDUs until receiving a number of responses (900).

- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT first rejecting, then accepting the connection request and maintaining the connection in the slave role.

The IUT reports the connection created with an HCI event.

The IUT maintains the connection using the Channel Selection Algorithm #1.

4.3.2.7 Extended Advertising, Accepting Connections

- Test Purpose

Tests that an advertiser IUT, using connectable advertising, receives a connection request on the secondary channel, stops advertising after the reception, and starts to maintain a connection in the slave role. The Lower Tester acts first in the initiating state, sending the connection request to the IUT on the secondary channel, and then starts to maintain a connection in the master role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.

- Reference

[10] 4.4.2.3, 4.3.2, 4.4.2.8, 4.4.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
• Test Procedure

For each round from 1 to 4

Repeat up to 20 Times

ADV EXT IND
(AdvMode: 01b, AuxPtr)
ADV EXT IND
(AdvMode: 01b, AuxPtr)
ADV EXT IND
(AdvMode: 01b, AuxPtr)
AUX ADV IND
(AdvMode: 01b, AdvA)
AUX CONNECT REQ
(InitA, AdvA)

AUX CONNECT_RSP
HCI LE Connection Complete Event
(Status: 0x00)

HCI LE Set Extended Advertising Parameters
(Extended Advertising, Connectable)

HCI Command Complete Event
(Status: 0x00)

HCI LE Set Extended Advertising_Enable
Enable

HCI Command Complete Event
(Status: 0x00)

HCI LE Advertising Set Terminated Event
(Status: 0x00)

HCI LE Channel Selection Algorithm Event
(Channel Selection Algorithm #2)

HCI Disconnect
(Reason: 0x13)

LL TERMINATE_IND
(ErrorCode: 0x13)

HCI Command Status Event
(Status: 0x00)

HCI Disconnection Complete Event
(Reason: 0x16)

Connection Interval

Data Packet

Data Packet

Data Packet

Data Packet

Continued in Part B...

Data Packet

Data Packet

Data Packet

Figure 4.129: Extended Advertising, Accepting Connections – Part A

Figure 4.130: Extended Advertising, Accepting Connections – Part B
For each round from 1–4 based on Table 4.41:

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. The Advertising_Event_Properties parameter shall be set to the value specified in Table 4.41 for this round. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set as specified in Table 4.42. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration). The Max_Extended_Advertising_Events[0] parameter is set to 0x00 (No Max).

3. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 01b with only the AuxPtr Extended Header field.

4. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 01b and with the AdvA Extended Header field present. The AdvA field shall contain the IUT’s Advertising Address. If this round used directed advertising, AUX_ADV_IND PDU shall also include the TargetA Extended Header field set to the Lower Tester’s Public Device Address. The Lower Tester responds with an AUX_CONNECT_REQ PDU on the secondary advertising channel T_IFS after receiving the AUX_ADV_IND PDU. The InitA field shall be set to the Lower Tester’s address. The AdvA field shall be set as shown in Table 4.41 for the current Round.

5. If the AUX_CONNECT_REQ PDU has an AdvA not equal to the IUT’s address, the Lower Tester receives no AUX_CONNECT_RSP packet from the IUT. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. Skip to step 15.

6. The Lower Tester receives an AUX_CONNECT_RSP PDU from the IUT on the secondary advertising channel T_IFS after sending the AUX_CONNECT_REQ. If no AUX_CONNECT_RSP PDU is received, repeat steps 3–6 up to 20 times. If the Lower Tester does not receive that PDU after 20 times, this test case ends with a Fail Verdict. After receiving an AUX_CONNECT_RSP PDU, verify that the IUT has started a connection by responding to the Lower Tester’s LL data packets.

7. The Lower Tester receives no further ADV_EXT_IND PDUs after the advertising interval from the IUT. The Lower Tester continues scanning for advertising packets during the free time while maintaining the connection in steps 10–13. If the Lower Tester receives a further advertising packet, the test ends with a Fail Verdict.

8. The Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

9. The Upper Tester receives an HCI_LE_Advertising_Set_Terminated event from the IUT with the Status set to 0x00, the Advertising_Handle from step 1, and the Connection_Handle from step 8.

10. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.

11. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

12. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

13. Repeat steps 10–12 for a number of events (100 events).

14. The Upper Tester terminates the connection.

15. Repeat steps 1–14 for each Round shown in Table 4.41.
<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Extended_Advertising_Parameters (Step 1)</th>
<th>AUX_CONNECT_REQ PDU (Step 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Advertising_Event_Properties</td>
<td>AdvA</td>
</tr>
<tr>
<td>1</td>
<td>0x0001</td>
<td>IUT</td>
</tr>
<tr>
<td>2</td>
<td>0x0001</td>
<td>Not IUT</td>
</tr>
<tr>
<td>3</td>
<td>0x0005</td>
<td>IUT</td>
</tr>
<tr>
<td>4</td>
<td>0x0005</td>
<td>Not IUT</td>
</tr>
</tbody>
</table>

*Table 4.41: Payload contents for each case variation*

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Advertising PHY</td>
</tr>
<tr>
<td><strong>4.3.2.7.1</strong> LL/CON/ADV/BV-05-C [Extended Advertising, Accepting Connections – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.3.2.7.2</strong> LL/CON/ADV/BV-12-C [Extended Advertising, Accepting Connections – LE 2M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.3.2.7.3</strong> LL/CON/ADV/BV-13-C [Extended Advertising, Accepting Connections – LE Coded PHY]</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

*Table 4.42: Extended Advertising, Accepting Connections Test Cases*

- **Expected Outcome**

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- If the advertising was directed, the AUX_ADV_IND has a TargetA field containing the Lower Tester’s address.
- The IUT responds to the AUX_CONNECT_REQ within the 2 µs range around T_IFS.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- If the AUX_CONNECT_REQ PDU has an AdvA not equal to the IUT’s address then the IUT shall not respond.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports termination of advertising with an HCI LE Advertising Set Terminated event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data on the expected PHY selected in step 1.

4.3.2.8 LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections]

- Test Purpose

Tests that an advertiser IUT, using undirected connectable advertising with legacy PDUs, receives a connection indication on the primary channel, stops advertising after the reception, and starts to maintain a connection in the slave role. The Lower Tester acts first in the initiating state, sending the connection request to the IUT on the primary channel, and then starts to maintain a connection in the master role. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.

- Reference

[10] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Connectable Undirected Advertising (selected Adv_INTERVAL_MIN, selected Adv_INTERVAL_MAX, supported type of own address, all supported advertising channels, Length of device name used, common device name)
Test Procedure

- **Lower Tester**
- **IUT**
- **Upper Tester**

- **HCI_LE_Set_Extended_Advertising_Parameters**
  - (00010011b Event, Public Address)
  - (Status: 0x00)
- **HCI_Command_Complete_Event**
- **HCI_LE_Set_Extended_Advertising_Enable**
  - (Enable)
  - (Status: 0x00)
- **HCI_Command_Complete_Event**
- **ADV_IND**
  - (AdvA)
- **ADV_IND**
  - (AdvA)
- **ADV_IND**
  - (AdvA)
- **CONNECT_IND**
  - (InitA, AdvA)
- **HCI_LE_Connection_Complete_Event**
  - (Status: 0x00)
- **HCI_LE_Channel_Selection_Algorithm_Event**
  - (Channel Selection Algorithm #2)
  - (Status: 0x00)
- **HCI_LE_Advertising_Set_Terminated_Event**
- **HCI_Disconnection_Complete_Event**
  - (Reason: 0x16)
  - (Status: 0x00)
- **HCI_Command_Status_Event**
  - (Status: 0x00)

Repeat up to 20 Times

Repeat 100 Times

Continued in Part B...

**Figure 4.131:** LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections – Part A]
1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. The Advertising_Event_Properties parameter shall be set to 00010011b (ADV_IND legacy PDU). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration). The Max_Extended_Advertising_Events[0] parameter is set to 0x00 (No Max).

3. The Lower Tester expects the IUT to send ADV_IND, with ChSel set to 1 (Channel Selection Algorithm #2), on the advertising channel. The AdvA field shall contain the IUT’s Advertising Address.

4. The Lower Tester responds with a CONNECT_IND PDU, with ChSel set to 1 (Channel Selection Algorithm #2), on the primary advertising channel. The InitA field shall be set to the Lower Tester’s address. The AdvA field shall be set to the IUT’s address.

5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). If the Upper Tester does not receive the event, repeat steps 3–5 up to 20 times. If the Upper Tester does not receive the event after 20 times, this test case ends with a Fail Verdict.

6. The Lower Tester continues scanning for advertising packets during the free time while maintaining the connection in steps 9–10. If the Lower Tester receives a further advertising packet, the test ends with a Fail Verdict.

7. Immediately after receiving the HCI_LE_Connection_Complete event, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

8. The Upper Tester receives an HCI_LE_Advertising_Set_Terminated event from the IUT with the Status set to 0x00, the Advertising_Handle from step 1, and the Connection_Handle from step 5.
9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
10. Repeat step 9 for a number of events (100 events).
11. The Upper Tester terminates the connection.
12. Repeat steps 1–4, except that in step 4 the AdvA field shall be set to an address other than the IUT’s address.
13. The Lower Tester expects the IUT to continue sending ADV_IND PDUs for at least 5 advertising intervals and responds with a CONNECT_IND PDU each time.

• Expected Outcome

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- In step 4, the IUT sends an ADV_IND, with ChSel set to 1, on the primary advertising channel using the LE 1M PHY.
- In step 5, the IUT reports the requested connection with an HCI LE Connection Complete event.
- In step 7, the IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- In step 8, the IUT reports termination of advertising with an HCI LE Advertising Set Terminated event.
- In step 9, the IUT sends and receives data using the LE 1M PHY.
- In step 13, if the CONNECT_IND PDU has an AdvA not equal to the IUT’s address then the IUT shall not respond.

4.3.2.9 LL/CON/ADV/BV-07-C [Accepting Connections, Channel Selection Algorithm #1]

• Test Purpose

Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates no support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to zero (0), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #1. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

• Reference

[10] 4.4.2.3, 4.3.2

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Connectable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command.
3. The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 1 on the advertising channel. The AdvA field shall contain the IUT’s Advertising Address.
4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 0. The InitA field shall be set to the Lower Tester’s address, and the AdvA field shall be set to the IUT’s Advertising Address.
5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4–5 up to 20 times or until the IUT stops advertising.
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester receives an
HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).

7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

10. Repeat steps 7–9 for a number of events (100 events).

• Expected Outcome

    **Pass Verdict**

    For all rounds described in the test procedure, the following condition shall occur:

    - The IUT sends ADV_IND PDUs with ChSel set to 1 on the primary advertising channel.
    - The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
    - The IUT reports the requested connection with an HCI LE Connection Complete event.
    - The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
    - The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

4.3.2.10 LL/CON/ADV/BV-08-C [Directed Advertising Connection, Channel Selection Algorithm #1]

• Test Purpose

Tests that an advertiser IUT receives a connection request to the directing advertising indication, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates no support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to zero (0), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #1. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

• Reference

[10] 4.4.2.4, 4.3.2

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Directed Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x01 (ADV_DIRECT_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address shall be set to the Lower Tester’s address. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command.

3. The Lower Tester expects the IUT to send ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT’s Advertising Address. The TargetA field shall contain the Lower Tester’s address.

4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 0. The InitA field shall be set to the Lower Tester’s address, and the AdvA field shall be set to the IUT’s Advertising Address.

5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4–5 up to 20 times or until the IUT stops advertising.
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).

7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

10. Repeat steps 7–9 for a number of events (100 events).

- Expected Outcome

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:
  - The IUT sends ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel.
  - The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
  - The IUT reports the requested connection with an HCI LE Connection Complete event.
  - The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
  - The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

**4.3.2.11 LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2]**

- **Test Purpose**

  Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #2. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- **Reference**

  [10] 4.4.2.3, 4.3.2

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State: Connectable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command.
3. The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT’s Advertising Address.
4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester’s address, and the AdvA field shall be set to the IUT’s Advertising Address.
5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #2. If no data packets are received, repeat steps 4–5 up to 20 times or until the IUT stops advertising.
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester receives an

Figure 4.135: LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2]
HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

10. Repeat steps 7–9 for a number of events (100 events).

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends ADV_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

4.3.2.12 LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2]

• Test Purpose

Tests that an advertiser IUT receives a connection request to the directing advertising indication, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #2. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

• Reference

[10] 4.4.2.4, 4.3.2

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Directed Advertising (selected Adv_INTERVAL_MIN, selected Adv_INTERVAL_MAX, supported type of own address, all supported advertising channels, Length of device name used, common device name)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x01 (ADV_DIRECT_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address shall be set to the Lower Tester's address. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command.

3. The Lower Tester expects the IUT to send ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT’s Advertising Address. The TargetA field shall contain the Lower Tester's address.

4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester’s address, and the AdvA field shall be set to the IUT’s Advertising Address.

5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s corrected formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #2. If no data packets are received, repeat steps 4–5 up to 20 times or until the IUT stops advertising.

Figure 4.136: LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2]
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

10. Repeat steps 7–9 for a number of events (100 events).

• Expected Outcome

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:
- The IUT sends ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

4.3.2.13 LL/CON/ADV/BI-01-C [Connection Supervision Timeout during Fail Connection Setup]

• Test Purpose

Verifies that an advertiser IUT correctly implements the connection supervision timeout, under a condition where it expires during connection setup.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role but with packets with invalid checksums.

• Reference

[3] 4.5.5, 4.5.9

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_IntervMax, selected Adv_IntervMin, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices) OR All White Listed (policy for advertiser))

• Test Procedure

Execute the test procedure using the common data channel selection parameters and the common connection interval to maintain the connection.
Lower Tester

IUT

Upper Tester

Undirected Advertising configured.

HCI_LE_Set_Advertising_Enable
(Enable)

ADV_IND

CONNECT_IND
(Channel)

HCI_Command_Complete_Event
(Status: 0x00)

REPEAT UNTIL CONNECTION SUPERVISION TIMER EXPIRES

Empty Data
(Incorrect CRC)

T_IFS

Empty Data
(SN, NESN)

HCI_Disconnection_Complete_Event
(Status: 0x08)

---

Figure 4.137: LL/CON/ADV/BI-01-C (Connection Supervision Timeout during fail connection setup)

1. Configure Lower Tester to initiate a connection and send packets with incorrect CRC after CONNECT_IND packet.
2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
3. Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only.
4. Lower Tester responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 4.
6. Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum, starting the first event using the common connection interval timing after the connection request and using the common data channel selection parameters.
7. Lower Tester receives a correctly formatted LL Data Channel PDU, not acknowledging the invalid packet, from the IUT, T_IFS after the PDU sent on the same data channel.
8. Repeat steps 6–7 until connection supervision timer expires.
9. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, with status parameter set to 'connection timeout' and with the connection handle matching to step 5.

**Expected Outcome**

Pass Verdict

The IUT responds to the master transmissions in at least 3 out of 6 events.
4.3.2.14 LL/CON/ADV/BV-11-C [Accepting Connections, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2]

- **Test Purpose**
  Tests that an advertiser IUT that only supports Channel Selection Algorithm #1 receives a connection request from a Lower Tester that supports Channel Selection Algorithm #2. The IUT stops advertising after the reception and starts to maintain a connection in the slave role with Channel Selection Algorithm #1. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role. The IUT observes the packets and timing from the Lower Tester to confirm the Lower Tester is using the Channel Selection Algorithm #1.

- **Reference**
  [10] 4.4.2.3, 4.3.2

- **Initial Condition**
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.
  State: Connectable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
• **Test Procedure**

![Diagram](image-url)

**Figure 4.138: LL/CON/ADV/BV-11-C [Accepting Connections, Channel Selection Algorithm #1]**

1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Advertising_Enable command.

3. The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 0 on the advertising channel. The AdvA field shall contain the IUT’s Advertising Address.

4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester’s address, and the AdvA field shall be set to the IUT’s Advertising Address.

5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester’s correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4–5 up to 20 times or until the IUT stops advertising.

6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.

10. Repeat steps 7–9 for a number of events (100 events).

- **Expected Outcome**

  **Pass Verdict**

  For all rounds described in the test procedure, the following condition shall occur:

  - The IUT sends ADV_IND PDUs with ChSel set to 0 on the primary advertising channel.
  - The IUT receives CONNECT_IND PDUs with ChSel set to 1.
  - The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
  - The IUT reports the requested connection with an HCI LE Connection Complete event.
  - The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

  **4.3.2.15 Extended Advertising, Accepting Connections with Random address**

- **Test Purpose**

  Tests that an advertiser IUT, using connectable advertising, receives a connection request on the secondary channel, stops advertising after the reception, and starts to maintain a connection in the slave role. The Lower Tester verifies this procedure with different types of random addresses used in the InitA field of the AUX_CONNECT_REQ.

- **Reference**

  [10] 4.4.2.3, 4.3.2, 4.4.2.8, 4.4.2

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State: Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all supported advertising channels, Length of device name used, common device name)
### Test Procedure

- **Repeat for LE 1M, LE Coded, and LE 2M PHYs (where supported):**
  - For each round from 1 to 3,
    - **HCI_LE_Set_Extended_Advertising_Parameters**
      - (Extended Advertising, Connectable)
      - **HCI_Command_Complete_Event** (Status: 0x00)
    - **HCI_LE_Set_Extended_Advertising_Enable**
      - **HCI_Command_Complete_Event** (Status: 0x00)

- **AUX_CONNECT_REQ**
  - (InitA, AdvA)

- **AUX_CONNECT_RSP**
  - (AdvMode: 01b, AdvA)

- **AUX_ADV_IND**
  - (AdvMode: 01b, AuxPtr)

- **ADV_EXT_IND**
  - (AdvMode: 01b, AuxPtr)

- **HCI_LE_Advertising_Set_Terminated_Event**
  - (Status: 0x00)

- **HCI_LE_Channel_Selection_Algorithm_Event**
  - (Channel Selection Algorithm #2)

- **Continued in Part B...**

---

**Figure 4.139:** Extended Advertising, Accepting Connections – Part A

**Figure 4.140:** Extended Advertising, Accepting Connections – Part B
For each round from 1–3 based on Table 4.43:

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0001 (Connectable advertising). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and the Secondary_Advertising_PHY shall be set to the PHY specified in Table 4.44. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).

2. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration). The Max_Extended_Advertising_Events[0] parameter is set to 0x00 (No Max).

3. The Lower Tester receives an ADV_EXT_IND packet from the IUT with AdvMode set to 01b with only the AuxPtr Extended Header field.

4. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 01b and with the AdvA Extended Header field present. The AdvA field shall contain the IUT’s Advertising Address. The Lower Tester responds with an AUX_CONNECT_REQ PDU on the secondary advertising channel T_IFS after receiving the AUX_ADV_IND PDU. The InitA field shall be set as shown in Table 4.43 for the current Round.

5. The Lower Tester receives an AUX_CONNECT_RSP PDU from the IUT on the secondary advertising channel T_IFS after sending the AUX_CONNECT_REQ with RxAdd = 1 (random address) and a valid TargetA address. If no AUX_CONNECT_RSP PDU is received, repeat steps 3–5 up to 20 times. If the Lower Tester does not receive that PDU after 20 times, this test case ends with a Fail Verdict. After receiving an AUX_CONNECT_RSP PDU, verify that the IUT has started to maintain a connection by responding to the Lower Tester’s LL data packets.

6. The Upper Tester receives an HCI_LE_Connection_Complete event from the IUT. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.

8. The Lower Tester receives a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.

9. The Upper Tester terminates the connection.

10. Repeat steps 1–9 for each Round shown in Table 4.43.

<table>
<thead>
<tr>
<th>Round</th>
<th>AUX_CONNECT_REQ PDU (Step 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>InitA</td>
</tr>
<tr>
<td>1</td>
<td>Random (static) address</td>
</tr>
<tr>
<td>2</td>
<td>Random (non-resolvable private) address</td>
</tr>
<tr>
<td>3</td>
<td>Random (resolvable private) address</td>
</tr>
</tbody>
</table>

Table 4.43: Payload contents for each case variation
• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.2.15.1 LL/CON/ADV/BV-14-C</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.2.15.2 LL/CON/ADV/BV-15-C</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.3.2.15.3 LL/CON/ADV/BV-16-C</td>
<td>LE Coded PHY</td>
</tr>
</tbody>
</table>

*Table 4.44: Test Case PHY Configuration*

• Expected Outcome

**Pass Verdict**

For all rounds described in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- The IUT responds to the AUX_CONNECT_REQ within the 2 µs range around T_IFS.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT sends and receives data on the expected PHY selected in step 1.
- The AUX_CONNECT_RSP uses correct address and address type values.

**Fail Verdict**

For all rounds described in the test procedure, the following condition occurred:

- The Lower Tester does not receive the PDU after 20 times in step 5.

**4.3.3 INI**

Tests that the IUT behaves according to the connection setup procedures as an initiator.

In all INI test cases where a connection gets created, the timing of the first packet from the master shall meet the requirements for clock accuracy. Specifically:

- Let t be the time, measured by the Lower Tester, from the end of the packet containing the CONNECT_IND or AUX_CONNECT_REQ PDU creating the connection to the start of the first data packet received from the master at the start of the connection.
- Let c be the connection event counter for that packet (normally c will equal 0, but if the Lower Tester misses the first packet from the IUT then it will be non-zero).
- Let a = transmitWindowDelay + transmitWindowOffset + connectionInterval * c.
- Let d = (masterSCA + slaveSCA) / 1000000.

Then t shall meet the requirement that:

\[ a * (1 - d) - 16 \leq t \leq (a + \text{transmitWindowSize}) * (1 + d) + 16 \]
The time intervals between connection events shall meet the requirements for clock accuracy. Specifically:

- Let \( t_i \) be the time of the \( i \)-th data packet received from the master at the start of a connection event (i.e. ignoring subsequent packets in connection events).

- Let \( c_i \) be the connection event counter for that packet.

- Let \( v_i = \text{connectionInterval} \times ((c_i - c_{i-1}) \mod 65536) \) be the nominal spacing between the relevant anchor points.

- Let \( d_i = (\text{masterSCA} + \text{slaveSCA}) / 1000000 \).

Then for every packet except the first, the timing of packet \( i \) shall meet the requirement that:

\[
v_i \times (1 - d_i) - 16 \leq t_i - t_{i-1} \leq v_i \times (1 + d_i) + 16
\]

Note: while the packets will normally be on consecutive connection events, this requirement allows for the case that one or more packets from the IUT are not received by the Lower Tester.

### 4.3.3.1 LL/CON/INI/BV-01-C [Connection Initiation]

- **Test Purpose**
  Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when initiating.

  The Lower Tester first acts in the advertising state, then accepts the connection and starts to maintain the IUT in the slave role, observing the packets and timing from the IUT.

- **Reference**
  [3] 1.3, 4.5.3, 4.5.4

- **Initial Condition**
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

  State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- **Test Procedure**
  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. Configure Lower Tester to use first supported advertising channel using a correct public address.
2. Upper Tester enables the initiator state in the IUT with the peer address and address type equal to the ones used by the Lower Tester. The Upper Tester receives an HCI_Command_Status_Event in response from the IUT with a Status of "success" (0x00).
3. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT random address and receives an HCI_Command_Complete event from the IUT with a Status of 0x0C.
4. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or step 5 executes.
5. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
6. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected.
7. After the CONNECT_IND packet has been received the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
8. Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

9. Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

10. Repeat a number of events (100 events) to conclude the timing accuracy.

11. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 6).

12. Configure Lower Tester to use first supported advertising channel and a public device address that differs from the IUT address in the most significant octet.

13. Repeat steps 2–11.

14. Configure Lower Tester to use the first supported advertising channel and a public device address that differs from the IUT address in the least significant octet.

15. Repeat steps 2–11.

16. Configure Lower Tester to use the first supported advertising channel and a public device address that differs from the IUT address in the most and least significant octets.

17. Repeat steps 2–11.

18. Configure Lower Tester to use second supported advertising channel using a correct public address.

19. Repeat steps 2–11.

20. Configure Lower Tester to use the second supported advertising channel and a public device address that differs from the IUT address in the most significant octet.

21. Repeat steps 2–11.

22. Configure Lower Tester to use the second supported advertising channel and a public device address that differs from the IUT address in the least significant octet.

23. Repeat steps 2–11.

24. Configure Lower Tester to use the second supported advertising channel and a public device address that differs from the IUT address in the most and least significant octets.

25. Repeat steps 2–11.

26. Configure Lower Tester to use the third supported advertising channel and a public device address that differs from the IUT address in the most significant octet.

27. Repeat steps 2–11.

28. Configure Lower Tester to use the third supported advertising channel and a public device address that differs from the IUT address in the least significant octet.

29. Repeat steps 2–11.

30. Configure Lower Tester to use the third supported advertising channel and a public device address that differs from the IUT address in the most and least significant octets.

31. Repeat steps 2–11.

• Expected Outcome

Pass Verdict

The test procedure completes using each advertising channel and device address with the IUT sending a connection request and maintaining the connection,

The IUT rejects the HCI_LE_Set_Random_Address command.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in the table below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 µs range around T_IFS.
The connection events’ time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events’ time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

The IUT maintains the connection using the Channel Selection Algorithm #1.

• Notes

The state 'Connected Master', which refers to the test procedure contents above, is used as an initial state in the connection handling tests.

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

4.3.3.2 LL/CON/INI/BV-02-C [Connecting to Directed Advertising]

• Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser using directed advertising events and starts to maintain a connection in the master role.

The Lower Tester first acts in the advertising state using directed advertising events, then accepts the connection and starts to maintain it in the slave role.

• Reference

[3] 4.5.4

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map
State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.

1. Configure Lower Tester as advertiser using all supported advertising channel with the interval between packets at most 0.5 ms and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends ADV_DIRECT_IND packets, each advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or step 4 executes.
4. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets.
5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.

6. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.

7. Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.

8. Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

9. Repeat a number of events (at least 1000 events) to conclude the timing accuracy.

10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 5).

- Expected Outcome

**Pass Verdict**

The test procedure completes with the IUT sending a connection request and maintaining the connection,

The connection events’ time intervals are within the range expressed for the sleep clock accuracy value,

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

The IUT maintains the connection using the Channel Selection Algorithm #1.

4.3.3.3 LL/CONINI/BV-03-C [Connection Initiation Missed Replies]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and after missing some reply transmissions from the slave, still manages to setup a connection in the master role. This test case reflects a typical scenario which the IUT must manage.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, and then begins to maintain the connection after omitting some events above the slave latency figure.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window and using the common device address. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel only using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or step 4 executes.
4. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
6. After the CONNECT_IND packet has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
7. Lower Tester does not reply to the correctly formatted LL Data Channel PDU and receives following correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals of minus (latest plus) maximum clock drift according to the drift rate indicated in the connection request, calculated for the connection interval used. Repeat for half the time required for the connection supervision timer to expire, then stop.
8. Lower Tester receives a correctly formatted LL Data Channel PDU and sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
9. Repeat a number of events (at least 10 events).
10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 5).
• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request, then continuing master transmissions until and after receiving the slave transmissions,

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

4.3.3.4 LL/CON/INI/BV-04-C [Connection Initiation Timeout]

• Test Purpose

Tests that an initiator IUT, after sending a connection request to an advertiser and missing reply transmissions from the slave until the connection supervision timer expires, considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, then does not start to maintain the connection, but observes the IUT reports in HCI events.

• Reference

[3] 4.5.4

• Initial Condition


State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, 0 slave latency, 100 ms timeout)

• Test Procedure

Execute the test procedure using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval, to maintain the connection. Use a zero slave latency AND a timeout parameter of 100 ms.
1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel only using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or step 4 executes.
4. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
5. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
6. After the CONNECT_IND packet has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel, in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
7. Lower Tester does not reply to the packet and receives correctly formatted LL Data Channel PDUs on subsequent data channels.
8. Repeat step 7 until the IUT stops.
9. Upper Tester receives an HCI_Disconnection_Complete event from the IUT with the reason parameter set to ‘connection failed to be established’ and with the connection handle matching to step 5.

- **Expected Outcome**
  
  **Pass Verdict**
  
  The IUT transmits a connection request, 
  
  The IUT then continues master transmissions until the connection supervision timer expires, 
  
  The IUT reports failure to establish a connection with an HCI event.

**4.3.3.5 LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected]**

- **Test Purpose**
  
  Tests that an initiator IUT sends connection requests correctly filtering advertiser devices. 
  
  The Lower Tester acts in the advertising state using undirected advertising events, observing the connection request packets from the IUT.

- **Reference**
  
  [3] 4.5.4, 4.3.4

- **Initial Condition**
  
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, 
  LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, 
  LL_initiator_Channel_Map, 
  
  State: Initiating (selected scan interval, selected scan window, white list is used, selected type of peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout) AND Connection Setup White Listed (Lower Tester address, selected type of peer address)
• Test Procedure

![Diagram of test procedure]

**Figure 4.145: LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected]**

1. Configure the White List of the IUT: public address type and a common device address.
2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a white listed device address and a random address type.
3. Upper Tester enables initiator state in the IUT using a public address type and the white listed device address.
4. Lower Tester sends ADV_IND packets each advertising event, using as advertising data the event sequence numbering and using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow.
5. Lower Tester receives no CONNECT_IND after any of the ADV_IND packets.
6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type.
7. Repeat steps 4–5.
8. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
9. Lower Tester sends ADV_IND packets, with event count as data encoded unsigned least significant bit first. Lower Tester sends the ADV_IND packet search advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or step 13 executes.
10. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
11. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower
    Tester address and connection interval selected.
12. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-
    encrypted, connection handle from step 4).
13. Repeat steps 1–12 changing public address type to random address type.

• Expected Outcome

  Pass Verdict
  The IUT sends a connection request to a white listed advertiser with the address and address type of
  the initiator in the advertising packet,
  The IUT does not send a connection request to a black listed advertiser with the address of the
  initiator in the advertising packet,
  The IUT reports the connection requested with an HCI event.
  The access address used by the IUT meets the requirements for access addresses.

4.3.3.6  LL/CON/INI/BV-07-C [Initiation Device Filtering: Directed]

• Test Purpose
  Tests that an initiator IUT sends connection requests correctly filtering advertiser devices.
  The Lower Tester acts in the advertising state using directed advertising events, observing the
  connection request packets from the IUT.

• Reference
  [3] 4.5.4, 4.3.4

• Initial Condition

  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
  LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
  LL_initiator_Channel_Map,

  State: Initiating (selected scan interval, selected scan window, white list is used, selected type of peer
  address, Lower Tester address, supported type of own address, common connection interval,
  common connection interval, common slave latency, common timeout) AND Connection Setup White
  Listed (Lower Tester address, selected type of peer address)
• Test Procedure

1. Configure the White List of the IUT: public address type and a common device address.
2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a white listed device address and a random address type.
3. Upper Tester enables initiator state in the IUT using a public address type and the white listed device address.
4. Lower Tester sends ADV_DIRECT_IND packets each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 \times \text{scanInterval} + 3 \times \text{scanWindow}$.
5. Lower Tester receives no CONNECT_IND after any of the ADV_DIRECT_IND packets.
6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type.
7. Repeat steps 4–5.
8. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
9. Lower Tester sends ADV_DIRECT_IND packets, each advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds $4 \times \text{scanInterval} + 3 \times \text{scanWindow}$ or step 13 executes.
10. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets.

11. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.

12. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 4).

13. Repeat steps 1–12 changing public address type to random address type.

- Expected Outcome

  **Pass Verdict**

  The IUT does not respond to connection requests from the Lower Tester in steps 4–8.

  The IUT sends a connection request to the white listed Lower Tester advertiser with the correct address and address type of the initiator in the advertising packet.

  The IUT reports the connection requested with an HCI event.

**4.3.3.7 LL/CON/INI/BV-08-C [Network Privacy – Connection Establishment responding to connectable undirected advertising, Initiator]**

- Test Purpose

  Verify that the IUT, when initiating connection establishment with the resolving list only containing a local IRK, the IUT uses the AdvA field received from the Lower Tester in the connectable undirected advertising event and generates a resolvable private address for the InitA field in the connect request packet.

- Reference

  [3] 1.3, 4.5.4

- Initial Condition

  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

  State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. Configure the Lower Tester to advertise using a valid public or static random address (identity address).
2. The Upper Tester adds the Identity Address of the Lower Tester to the Resolving List with all zero peer IRK, and with a local IRK.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel, using the selected advertising interval.
5. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets. The InitA field contains a resolvable private address from the IUT.
6. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
7. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request.
The IUT reports the connection setup with the HCI event.
The InitA field in the connection request uses a properly generated resolvable private address.

4.3.3.8 LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator]

• Test Purpose

Verify that the IUT when initiating connection establishment only connects to devices that are in the resolving list. The Lower Tester uses connectable undirected advertising.

• Reference

[3] 1.3, 4.5.4

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using a resolvable private address in the AdvA field of the advertising packets.
The Lower Tester has previously distributed its IRK to the IUT.
Test Procedure

1. Configure the Lower Tester to start advertising with a resolvable private address generated from a random IRK.
2. The Upper Tester adds the Lower Tester to the resolving list using a different IRK than in step 1.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow.
5. The IUT compares the address by checking against its resolving list and does not find a match.

Figure 4.148: LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator]
6. The Lower Tester receives no CONNECT_IND after any of the ADV_IND packets.
7. The Lower Tester stops advertising.
8. The Lower Tester begins advertising again using the correct resolvable address, which matches the one in the IUT resolving list. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow, or step 9 occurs.
9. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
10. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester's RPA and Identity address and connection interval selected.
11. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
13. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
15. The Upper Tester terminates the connection.

- Expected Outcome

**Pass Verdict**
The IUT receives and ignores connectable advertising with a resolvable private address which is not in the resolving list.
The IUT then receives a resolvable private address in the AdvA field from the Lower Tester which is in the resolving list.
The test procedure completes with the IUT sending a connection request and maintaining the connection.
The IUT reports the connection setup with an HCI event.

4.3.3.9 LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator]

- Test Purpose
Verify that the IUT when initiating connection establishment with the resolving list connects only to peer devices that are in the resolving list. The Lower Tester uses directed advertising. The IUT may use a public or static random device address.

- Reference
[3] 1.3, 4.5.4

- Initial Condition
Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map
State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)
Test Procedure

Figure 4.149: LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator]
1. Configure the Lower Tester to use the first supported advertising channel and a valid resolvable private address in the AdvA field, which is not known by the IUT (not in the resolving list).
2. The Upper Tester adds the Lower Tester to the resolving list using a different IRK than in step 1.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.
5. The IUT resolves and compares the address in the AdvA field by checking against its resolving list and does not find a match.
6. Repeat the advertising steps 4–5 for at least 20 advertising intervals.
7. The Lower Tester stops advertising.
8. The Lower Tester begins advertising again using the correct resolvable address, which matches the one stored in the IUT’s resolving list.
9. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets. The AdvA field is the same as the one received in the ADV_DIRECT_IND.
10. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
11. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
13. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
15. The Upper Tester terminates the connection.

- **Expected Outcome**

  **Pass Verdict**

  The IUT receives directed advertising with a resolvable private address in the AdvA field from the Lower Tester.

  The first AdvA address used does not match the resolving list and is ignored.

  The 2nd AdvA address used matches the resolving list and is accepted.

  The IUT uses the identity address in the InitA field of the connect request packet.

  The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

### 4.3.3.10 LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator]

- **Test Purpose**

  Verify that the IUT when initiating connection establishment with the resolving list connects only to directed advertisements that are addressed to the IUT.

- **Reference**

  [3] 1.3, 4.5.4

- **Initial Condition**

  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map
State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- **Test Procedure**

![Test Diagram]

Figure 4.150: LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator]
1. Configure the Lower Tester to start directed advertising using a valid resolvable private address in the AdvA field.
2. The Upper Tester adds the Lower Tester to the resolving list with both peer Identity (IRK and Identity Address) and local IRK.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_DIRECT_IND packets, each advertising event, using the selected advertising interval and with a resolvable private address in the InitA field generated from a random IRK different from the one distributed to the IUT.
5. The IUT tries to resolve the address in the InitA field by checking against its resolving list and does not find a match.
6. Repeat the advertising steps 4–5 for at least 20 advertising intervals.
7. The Lower Tester stops advertising.
8. The Lower Tester begins advertising again using the correct resolvable address, which resolves with the IUT’s local IRK.
9. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets. The InitA field contains a resolvable private address generated by the IUT. The address should be different from the address received in the ADV_DIRECT_IND packet.
10. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
11. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
13. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
15. The IUT (master) maintains the link for the address refresh timeout before terminating the connection.
16. Repeat steps 3–10 to see that the InitA field of the CONNECT_IND have been refreshed.
17. The Upper Tester terminates the connection.

• Expected Outcome

Pass Verdict

The IUT receives directed advertising with a resolvable private address in the AdvA field and an invalid resolvable private address in the InitA field from the Lower Tester.

The first InitA address used does not match the resolving list and is ignored.

The second InitA address used matches the resolving list and is accepted.

The IUT sends a resolvable private address in the InitA field of the connect request packet.

The IUT address is verified against the resolving list.

The Upper Tester then disconnects after the address refresh timeout has passed and reconnects to see that the IUT generates a new resolvable private address.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.
4.3.3.11 LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator]

- **Test Purpose**
  Verify the IUT when initiating private connection establishment with the resolving list does not connect to directed advertisements that are addressed to the IUT using its identity address.

- **Reference**
  [3] 1.3, 4.5.4

- **Initial Condition**
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map
  State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)
• Test Procedure

Figure 4.151: LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator]
1. The Upper Tester adds the address of the Lower Tester to the resolving list with the peer IRK.
2. The Upper Tester enables the initiator state in the IUT using a private address.
3. Configure the Lower Tester to start directed advertising using a valid resolvable private address in the AdvA field and the identity address of the IUT (either the public address or the static random address configured in the initial conditions).
4. Lower Tester sends ADV_DIRECT_IND packets, each advertising event, using the selected advertising interval and with public address of the IUT in the InitA.
5. The IUT sees its own public address in the ADV_DIRECT_IND, but ignores it since it is initiating using a private address.
6. Repeat steps 4–5 for at least 20 advertising intervals.
7. The Lower Tester starts advertising again using a resolvable private address generated using the IUT IRK in the InitA field.
8. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets. The InitA field contains a resolvable private address generated by the IUT. The address should be different from the address received in the ADV_DIRECT_IND packet.
9. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
10. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
11. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
12. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
13. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
14. The IUT (master) maintains the link for the address refresh timeout before terminating the connection.
15. Repeat steps 3–9 to see that the InitA field of the CONNECT_IND have been refreshed.
16. The Upper Tester terminates the connection.

- Expected Outcome

Pass Verdict

The IUT receives directed advertising with a resolvable private address in the AdvA field and the identity address in the InitA field from the Lower Tester.

The first InitA address used does not match the own_address_type to use only Resolvable Private Address and is ignored.

The second InitA address used matches the resolving list and is accepted.

The IUT sends a resolvable private address in the InitA field of the connect request.

The IUT address is verified against the resolving list.

The Upper Tester then disconnects after the address refresh timeout has passed and reconnects to see that the IUT generates a new resolvable private address.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

4.3.3.12 Extended Scanning, Connection Initiation

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser on the secondary advertising channel and starts to maintain a connection in the master role. The Lower Tester first acts in the
extended advertising state, then accepts the connection and starts to maintain the IUT in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- **Reference**
  
  [10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- **Initial Condition**


  State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- **Test Procedure**

  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. The Upper Tester sends an HCI_LE_Extended_Create_Connection command to the IUT. The Initiating_PHYs parameter shall be set as specified in Table 4.45, Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Initiating_Filter_Policy shall be set to 0x00 (use Peer_Address_Type and Peer_Address), and Own_Address_Type shall be set to 0x00 (Public Device Address). The peer address and address type shall be set to the ones used by the Lower Tester. The Upper Tester receives an HCI_Command_Status event in response.

2. The Lower Tester begins advertising using the ADV_EXT_IND PDU on the PHY as specified in Table 4.45 with the AuxPtr field referencing the AUX_ADV_IND.

3. The Lower Tester receives an AUX_CONNECT_REQ PDU on the secondary advertising channel as specified in Table 4.45 T_IFS after sending any of the AUX_ADV_IND PDUs.

4. The Lower Tester sends an AUX_CONNECT_RSP PDU from the IUT on the secondary advertising channel as specified in Table 4.45 T_IFS after receiving the AUX_CONNECT_REQ.
5. The Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

6. After the AUX_CONNECT_RSP has been sent, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from AUX_CONNECT_REQ.

7. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

8. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

9. Repeat a number of events (100 events) to conclude the timing accuracy.

10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 5).

11. Repeat steps 1–10, except instead of step 4, the Lower Tester does not respond to the first AUX_CONNECT_REQ PDU, continues advertising, and expects the IUT to retry the connection request following the backoff algorithm by sending a second AUX_CONNECT_REQ PDU on the secondary advertising channel on the PHY as specified in Table 4.45 T_IFS after any of the AUX_ADV_IND PDUs.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initiating PHY</td>
</tr>
<tr>
<td>4.3.3.12.1</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.3.3.12.2</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.3.3.12.3</td>
<td>0x04 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

Table 4.45: Extended Scanning, Connection Initiation Test Cases

• Expected Outcome

Pass Verdict
The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.46 below.

The number of timing measurements for event starts from the IUT is at least 100.
The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS. The connection events’ time intervals are within the range expressed for the sleep clock accuracy value. The difference between the sum of the measured connection events’ time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value. The IUT reports the connection setup with the HCI event. The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event. When a connection request receives no response, the IUT retries according to the backoff algorithm. The access address used by the IUT meets the requirements for access addresses.

• Notes
Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

*Table 4.46: Timing requirements*

4.3.3.13 LL/CON/INI/BV-14-C [Connection Initiation, Channel Selection Algorithm #1]

• Test Purpose
Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates no support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to zero (0), then accepts the connection and starts to maintain it in the slave role, observing the packet and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #1 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

• Reference
[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4
• Initial Condition


State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

• Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.

1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to 0x00 (use Peer_Address_Type and Peer_Address). The Upper Tester receives an HCI_Command_Status event in response.
2. The Lower Tester begins advertising using ADV_IND PDUs with ChSel set to 0.
3. The Lower Tester receives a CONNECT_IND PDU with ChSel set to 0 or 1 T_IFS after sending any of the ADV_IND PDUs.

4. The Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).

5. The Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.

6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

7. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

8. Repeat a number of events (100 events) to conclude the timing accuracy.

9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 4).

• Expected Outcome

Pass Verdict
The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.47 below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS.

The connection events’ time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events’ time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses.

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

• Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
</tbody>
</table>
### Table 4.47: Timing requirements.

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

#### 4.3.3.14 LL/CON/INI/BV-15-C [Connecting to Directed Advertising, Channel Selection Algorithm #1]

- **Test Purpose**
  Tests that an initiator IUT sends a connection request to an advertiser using directed advertising and starts to maintain a connection in the master role when the advertisement indicates no support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to zero (0), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #1 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- **Reference**
  [10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- **Initial Condition**

  State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- **Test Procedure**
  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to 0x00 (use Peer_Address_Type and Peer_Address). The Upper Tester receives an HCI_Command_Status event in response.

2. The Lower Tester begins advertising using ADV_DIRECT_IND PDUs with ChSel set to 0.

3. The Lower Tester receives a CONNECT_IND PDU with ChSel set to 0 or 1 T_IFS after sending any of the ADV_DIRECT_IND PDUs.

4. The Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).

5. The Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.

6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

7. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from step 4).

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.48 below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses.

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

• Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

Table 4.48: Timing requirements
4.3.3.15 LL/CON/INI/BV-16-C [Connection Initiation, Channel Selection Algorithm #2]

- **Test Purpose**
  Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- **Reference**
  [10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- **Initial Condition**

  State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- **Test Procedure**
  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to 0x00 (use Peer_Address_Type and Peer_Address). The Upper Tester receives an HCI_Command_Status event in response.
2. The Lower Tester begins advertising using ADV_IND PDUs with ChSel set to 1.
3. The Lower Tester receives a CONNECT_IND PDU with ChSel set to 1 T_IFS after sending any of the ADV_IND PDUs.
4. The Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
5. The Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
7. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 4).

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.49 below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses.

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

• Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

Table 4.49: Timing requirements.
4.3.3.16  LL/CON/INI/BV-17-C [Connecting to Directed Advertising, Channel Selection Algorithm #2]

- Test Purpose
  Tests that an initiator IUT sends a connection request to an advertiser using directed advertising and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference
  [10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- Initial Condition
  State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure
  Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.
1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to 0x00 (use Peer(Address_Type and Peer_Address). The Upper Tester receives an HCI_Command_Status event in response.

2. The Lower Tester begins advertising using ADV_DIRECT_IND PDUs with ChSel set to 1.

3. The Lower Tester receives a CONNECT_IND PDU with ChSel set to 1 T_IFS after sending any of the ADV_DIRECT_IND PDUs.

4. The Upper Tester receives an HCI_LE_Connection_Complete_Event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester receives an HCI_LE_Channel_Selection_Algorithm_Event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).

5. The Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.

6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

7. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 4).

- **Expected Outcome**

**Pass Verdict**

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.50 below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses.

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

- **Notes**

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

<table>
<thead>
<tr>
<th>LL_SCA</th>
<th>Event time interval accepted</th>
<th>Limit for drift accepted / 100 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ppm</td>
<td>29.98 ms to 30.02 ms</td>
<td>1.50 ms</td>
</tr>
<tr>
<td>250 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.75 ms</td>
</tr>
<tr>
<td>150 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.45 ms</td>
</tr>
<tr>
<td>100 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.30 ms</td>
</tr>
<tr>
<td>75 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.23 ms</td>
</tr>
<tr>
<td>50 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.15 ms</td>
</tr>
<tr>
<td>30 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.09 ms</td>
</tr>
<tr>
<td>20 ppm</td>
<td>29.99 ms to 30.01 ms</td>
<td>0.06 ms</td>
</tr>
</tbody>
</table>

*Table 4.50: Timing requirements*
### 4.3.3.17 Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address

- **Test Purpose**
  Verify that the IUT when initiating connection establishment does not connect to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode.

- **Reference**
  [3] 1.3, 4.5.4

- **Initial Condition**
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

  State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

  The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.

  The Lower Tester has previously distributed its IRK to the IUT.

  The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.

- **Test Procedure**

  ![Diagram](image)

  *Figure 4.157: Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address*
1. Configure the Lower Tester to start advertising with its device identity address.
2. The Upper Tester adds the Lower Tester to the resolving list using an IRK.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow.
5. The IUT compares the address by checking against its resolving list and finds a match with network privacy mode.
6. The Lower Tester receives no CONNECT_IND after any of the ADV_IND packets.
7. The Lower Tester stops advertising.

**Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Address Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.3.17.1 LL/CON/INI/BV-18-C [Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address]</td>
<td>Enabled</td>
</tr>
<tr>
<td>4.3.3.17.2 LL/CON/INI/BV-24-C [Network Privacy - Connection Establishment using resolving list, Initiator, Ignore Identity Address, with address resolution disabled]</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Table 4.51: Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address Test Cases

**Expected Outcome**

Pass Verdict

The IUT receives and ignores connectable advertising with a device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode.

4.3.3.18 LL/CON/INI/BV-19-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Ignore Identity Address]

**Test Purpose**

Verify that the IUT when initiating connection establishment does not connect to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode. The Lower Tester uses directed advertising. The IUT may use a public or static random device address.

**Reference**

[3] 1.3, 4.5.4

**Initial Condition**

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)
The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.

- **Test Procedure**

  1. Configure the Lower Tester to start directed advertising with its device identity address.
  2. The Upper Tester adds the Lower Tester to the resolving list using an IRK.
  3. The Upper Tester enables the initiator state in the IUT.
  4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.
  5. The IUT resolves and compares the address in the AdvA field by checking against its resolving list and finds a match with network privacy mode.
  6. The Lower Tester receives no CONNECT_IND after any of the ADV_DIRECT_IND packets.
  7. Repeat the advertising steps 4–5 for at least 20 advertising intervals.
  8. The Lower Tester stops advertising.

- **Expected Outcome**

  **Pass Verdict**

  The IUT receives and ignores connectable advertising with a device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode.
4.3.3.19 LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address]

- **Test Purpose**
  Verify that the IUT when initiating connection establishment connects to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode.

- **Reference**
  [3] 1.3, 4.5.4

- **Initial Condition**
  Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map
  State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)
  The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.
  The Lower Tester has previously distributed its IRK to the IUT.
Figure 4.159: LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address]

1. Configure the Lower Tester to start advertising with its device identity address.
2. The Upper Tester adds the Lower Tester to the resolving list using an IRK and sets the entry to device privacy mode.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow, or step 5 occurs.
5. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
6. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT
   including the Lower Tester's Identity address and connection interval selected.
7. After the CONNECT_IND has been received, the Lower Tester receives the first correctly
   formatted LL Data Channel PDU on the data channel.
8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data
   channel using the acknowledgement scheme.
9. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data
   channels at connection intervals, calculated for the connection interval used.
10. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
11. The Upper Tester commands the IUT (master) to terminate the connection.

• Expected Outcome

Pass Verdict

The IUT receives and creates a connection for connectable advertising with a device identity address
when the identity address and an associated IRK are in the resolving list using device privacy mode.

The test procedure completes with the IUT sending a connection request and maintaining the
connection.

The IUT reports the connection setup with an HCI event.

4.3.3.20 LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed
advertising and resolving list, Initiator, Accept Identity Address]

• Test Purpose

Verify that the IUT when initiating connection establishment connects to a device advertising using its
device identity address when the identity address and an associated IRK are in the resolving list
using device privacy mode. The Lower Tester uses directed advertising. The IUT may use a public or
static random device address.

• Reference

[3] 1.3, 4.5.4

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer
address, Lower Tester address, supported type of own address, common connection interval,
common connection interval, common slave latency, common timeout)
• Test Procedure

Lower Tester → IUT → Upper Tester

- **HCI_LE_Create_Connection**
  
  (Init_filter_pol=0, Own_addr_type=0x2, Peer_Addr_Type=0x0, ... Peer_addr, Device Privacy Mode)

  → **HCI_Command_Complete_Event**
  
  (Status: 0x00)

- **HCI_LE_Set_Privacy_Mode**
  
  (Peer_addr_type, Peer_addr, Device Privacy Mode)

  → **HCI_Command_Complete_Event**
  
  (Status: 0x00)

- **HCI_LE_Set_Resolvable_Private_Address_Timeout**
  
  (RPA_Timeout)

  → **HCI_Command_Complete_Event**
  
  (Status: 0x00)

- **HCI_LE_Set_Address_Resolution_Enable**
  
  (Enable)

  → **HCI_Command_Complete_Event**
  
  (Status: 0x00)

- **HCI_LE_Create_Connection**
  
  (Init_filter_pol=0, Own_addr_type=0x2, Peer_Addr_Type=0x0, Peer_addr)

  → **HCI_Command_Status_Event**
  
  (Status: 0x00)

- **ADV_DIRECT_IND**
  
  (Identity Address)

- **CONNECT_IND**
  
  (Identity Address)

- **HCI_LE_Enhanced_Connection_Complete_Event**
  
  (Status: 0x00)

- **Empty Data**
  
  REPEAT 100 TIMES

Figure 4.160: LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Accept Identity Address]
1. Configure the Lower Tester to start directed advertising with its device identity address.
2. The Upper Tester adds the Lower Tester to the resolving list using an IRK and sets the entry to device privacy mode.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.
5. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets. The AdvA field is the same as the one received in the ADV_DIRECT_IND.
6. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
7. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
9. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
10. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
11. The Upper Tester commands the IUT (master) to terminate the connection.

• Expected Outcome

Pass Verdict

The IUT receives and creates a connection for directed advertising with a device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

4.3.3.21 LL/CON/INI/BI-01-C [Connection Initiation Invalid CRC]

• Test Purpose

Tests that an initiator IUT ignores advertising packets with an invalid checksum.

The Lower Tester acts in the advertising state transmitting packets with invalid checksums and observes whether the IUT makes a connection request, then stops the attempt to create a connection.

• Reference

[3] 4.5.4

• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map,

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, any connection interval, any connection interval, any slave latency, any timeout).

• Test Procedure

Execute the test procedure advertising using a selected scan interval and window once using a common public device address.
1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends an ADV_IND packet with an invalid checksum, each advertising event using the selected advertising channel only, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow.
4. Lower Tester receives no CONNECT_IND packet T_IFS after any of the ADV_IND packets.
5. Upper Tester sends an HCI_LE_Create_Connection_Cancel command to the IUT to stop the connection setup and receives an HCI_Command_Complete event in response.

- Expected Outcome

**Pass Verdict**

The test procedure completes without the IUT sending a connection request and stopping the connection setup procedure reporting it with the HCI event.

### 4.3.3.22 LL/CON/INI/BI-02-C [Slave Packets Invalid CRC]

- **Test Purpose**

Tests that an initiator IUT sends a connection request to an advertiser and receiving reply transmissions with invalid checksums from the slave up to the point of expiring the connection supervision timer, considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, and then begins to maintain the connection in the slave role but with packets with an invalid checksum.

- **Reference**

[3] 4.5.4, 4.51
• **Initial Condition**


State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, 0 slave latency, 100 ms timeout).

• **Test Procedure**

Execute the test procedure using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval to maintain the connection. Use slave latency = 0 and timeout parameter of 100 ms.

![Diagram of LL/CON/INI/BI-02-C](image)

**Figure 4.162: LL/CON/INI/BI-02-C [Slave packets Invalid CRC]**

1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends an ADV_IND packet with the selected address and 1 byte as data, each advertising event on the selected advertising channel only, using the selected advertising interval. Lower Tester repeats until the time exceeds 4 * scanInterval + 3 * scanWindow or 4 executions. Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.

4. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.

5. After the CONNECT_IND packet has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel in the range of transmitWindowOffset and transmitWindowSize.

6. Lower Tester
   a) Sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
      Receives a correctly formatted LL Data Channel PDU on the same data channel inside the same connection event. If this second Data Channel PDU in the connection event is received, Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
   b) Receives correctly formatted LL Data Channel PDU on subsequent data channel in the next connection interval of minus (latest plus) maximum clock drift according to the drift rate indicated in the connection request, calculated for the connection interval used.
   c) Sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
   d) Receives a correctly formatted LL Data Channel PDU on the same connection event. If this second Data Channel PDU in the connect event is received, Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel using the acknowledgement scheme.

7. Repeat steps 6b–6c until IUT stops because of supervision timeout.

8. OR

9. Repeat steps 6b–6d until IUT stops because of supervision timeout.

10. Upper Tester receives an HCI_Discconnection_Complete event from the IUT, indicating 'CONNECTION TIMEOUT' and with the connection handle matching step 6.

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request, then continuing master transmissions until Connection Supervision expires.

The IUT reports the connection timeout with an HCI event.

4.3.3.23 LL/CON/INI/BV-22-C [Connection Initiation, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2]

• Test Purpose

Tests that an initiator IUT that only supports Channel Selection Algorithm #1 sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packet and timing from the IUT. The IUT confirms the Channel Selection Algorithm #1 is used for the connection.

• Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4
• Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer
address, lower tester address, supported type of own address, common connection interval, common
connection interval, common slave latency, common timeout)

• Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data
channel selection parameters and the common connection interval, latency and timeout to maintain
the connection.

Figure 4.163: LL/CON/INI/BV-22-C [Connection Initiation, Channel Selection Algorithm #1]

1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and
address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to
0x00 (use Peer_Address_Type and Peer_Address). The Upper Tester receives an
HCI_Command_Status event in response.
2. The Lower Tester begins advertising using ADV_IND PDUs with ChSel set to 1.
3. The Lower Tester receives a CONNECT_IND PDU with ChSel set to 0 T_IFS after sending any
of the ADV_IND PDUs.
4. The Upper Tester receives an HCI_LE_Connection_Complete_event from the IUT including the
lower tester address and the connection interval selected.
5. The Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.

6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

7. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

8. Repeat a number of events (100 events) to conclude the timing accuracy.

9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 4).

• Expected Outcome

   Pass Verdict

   The test procedure completes with the IUT sending a connection request and maintaining the connection.

   The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in Table 4.46 above.

   The number of timing measurements for event starts from the IUT is at least 100.

   The timing deviations detected for packets in active mode are within the 2 μs range around T_IFS.

   The connection events’ time intervals are within the range expressed for the sleep clock accuracy value.

   The difference between the sum of the measured connection events’ time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

   The IUT reports the connection setup with the HCI event.

   The access address used by the IUT meets the requirements for access addresses.

   The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

• Notes

   Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted.

   **4.3.3.24 LL/CON/INI/BV-23-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled]**

• Test Purpose

   Verify that the IUT when initiating connection establishment using Whitelist connects to a device advertising using its device identity address when the peer address is in the resolving list with valid local IRK and address resolution is disabled.

• Reference

   [3] 6.4
• **Initial Condition**

  Parameters: LL\_initiator\_scanInterval\_MIN, LL\_initiator\_scanInterval\_MAX, LL\_initiator\_scanWindow\_MIN, LL\_initiator\_scanWindow\_MAX, LL\_initiator\_Adv\_Channel\_Map, LL\_initiator\_Channel\_Map

  State: Initiating (selected scan interval, selected scan window, White List is used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

  The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.

  The Peer IRK for IUT is set to zeros.

• **Test Procedure**

![Diagram](image)

_Figure 4.164_ LL/CON/INI/BV-23-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled]
1. Configure the Lower Tester to start advertising with its device identity address.
2. The Upper Tester populates the resolving list with the Lower Tester’s address. The Upper Tester receives HCI_Command_Complete with status Success (0x00).
3. The Upper Tester adds 2 more resolving list entries not equal to the Lower Tester with different local IRK for each entry. The Upper Tester receives HCI_Command_Complete with status Success (0x00) or Memory Capacity Exceeded (0x07).
4. The Upper Tester adds the Lower Tester’s address to the white list along with 2 other whitelist entries.
5. The upper tester issues HCI_LE_Create_Connection command to the IUT with own address type set to 0x02, Initiator Filter policy set to 0x01 and peer address set to the lower tester address.
6. Lower Tester sends adv packets, each advertising event on the selected advertising channel, using the selected advertising interval.
7. IUT receives the adv packet, and sends connect request with own address set to RPA, generated from the resolving list populated for the peer address.
8. The Lower Tester receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets. The AdvA field is the same as the one received in the ADV_IND and INITA as RPA.
9. Lower Tester resolves private address received from the IUT using assigned IRK.
10. Upper Tester receives an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
11. After the CONNECT_IND has been received, the Lower Tester receives the first correctly formatted LL Data Channel PDU on the data channel.
12. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
13. The Upper Tester terminates the connection.

**Expected Outcome**

**Pass Verdict**

- IUT sends a Connection Indication PDU to Lower Tester with Own Address set to RPA upon receiving an ADV_IND PDU from the Lower Tester advertising with Identity Address when IUT contains a matching entry for the Lower Tester Address in the Resolving List and Address Resolution is disabled at IUT.

- The test procedure completes with the IUT sending a connection request with RPA and maintaining the connection.

- The IUT reports the connection setup with an HCI event.

### 4.3.4 SLA

Tests that the IUT behaves according to the connection procedures as a slave.

#### 4.3.4.1 LL/CON/SLA/BV-02-C [Slave Asymmetric Connections]

**Test Purpose**

Tests that a slave IUT responds in all events in a connection where the slave latency parameter is not zero.

The Lower Tester acts in the master role in the connection and observes the slave packet timing and packet contents on the data channels in use.

**Reference**

[3] 4.5
• Initial Condition

Parameters: LL_slave_connSlaveLatency_MIN

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

• Test Procedure

Execute the test procedure using the common connection parameters, with slave latency of 5 and a timeout value of 200 ms.

1. Upper Tester sends HCI_LE_Read_Buffer_Size and receives an HCI_command_complete event.
2. Optional: Upper Tester sends HCI_Read_Buffer_Size and receives an HCI_command_complete event.
3. Lower Tester sends a DATA packet once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received matches the next SN and by using the next NESN where the SN in the previous packet correctly received matches the current NESN.
4. Upper Tester receives an HCI_LE_Data_Packet from the IUT containing a data element sent in 3 and with the Packet_Boundary_Flag flag set.
5. Lower Tester receives an empty data packet T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Allow mismatches for packets not received correctly.
6. Repeat steps 3–5 30 times.
7. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
• Test Condition
  The parameters in this test are calculated for a BER of 0.1 percent or better.

• Expected Outcome
  **Pass Verdict**
  The IUT responds to Lower Tester according to latency value (at least 5 packets in 30).

• Notes
  The error rate calculation considers only the access address resulting in that roughly 6 percent of
  symmetric events can be missed by a slave device.

4.3.4.2  LL/CON/SLA/BV-04-C [Slave Sending Data]

• Test Purpose
  Tests that a slave IUT is able to send data to a master device.
  The Lower Tester acts in the master role in the connection, submits data from host of the IUT to
  transmit and using the acknowledgement scheme, observes the data in the packets from the IUT.

• Reference
  [3] 4.5

• Initial Condition
  **Parameters:** LL_slave_payload_length_MIN, LL_slave_payload_length_MAX
  **State:** Connected Slave (any advertising interval, any advertising interval, public address, any
  advertising channel map, common connection interval, 0 slave latency, common timeout, common
  channel map, any SCA value)

• Test Procedure
  Execute the test procedure using the connection handle and data packet length from the execution of
  the preamble steps.
1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted. It receives the appropriate HCI_Number_Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.

2. Lower Tester receives a DATA packet from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1.
3. Lower Tester sends an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
4. Repeat steps 2 and 3 until all elements submitted in step 1 have been received.
5. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x01 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) is successfully submitted. It receives the appropriate HCI_Number_Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
6. Lower Tester receives a DATA packet from the IUT, with the LLID field set to 0x01 and containing a data element submitted in step 5. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
7. Repeat step 6 until all elements submitted in step 5 have been received.
8. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, in the following packets with the Packet_Boundary_Flag flag equal to 0x01 until the selected number of octets (1000) are successfully submitted. It receives the appropriate HCI_Number_Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
9. Lower Tester receives a DATA packet from the IUT, containing a data element submitted in step 8: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following packets. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
10. Repeat step 9 until all data sent in step 8 have been reported.
11. If the devices have a data_packet_length (as defined in Figure 4.4: Buffer Size Read Preamble Steps) less than or equal to 27 octets, skip to step 14; otherwise continue with step 12.
12. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, using a data total length generated randomly within the interval from 28 to the data_packet_length (as defined in Figure 4.4: Buffer Size Read Preamble Steps), until the selected number of octets (1000) are successfully submitted.
13. For each HCI Data packet command sent by the Upper Tester in step 12, the Lower Tester receives two or more DATA packet from the IUT, with the LLID field set to 0x02 in the first packet with data and set to 0x01 in the following packets, until all the data elements sent to the IUT in step 12 have been received from the IUT. The Lower Tester sends an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
14. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble step execution).

• Expected Outcome
  Pass Verdict
  The test procedure completes with the IUT sending all of the data.
  The IUT maintains correct sequence of the fragmentation flags as specified in Section 4.1.6.

4.3.4.3 LL/CON/SLA/BV-05-C [Slave Receiving Data]
Tests that a slave IUT is able to receive data from a master device.

The Lower Tester acts in the master role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.
• Reference

[3] 4.5

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value).

• Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.

1. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.

2. Lower Tester sends a DATA packet once a connection interval to the IUT, using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00. Lower Tester receives a DATA packet in response from the IUT.
3. Repeat step 2 until all data sent in step 1 has been reported.
4. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 1 with the Packet_Boundary_Flag flag set to 0x02.
5. Configure Lower Tester to send 100 data packets with the LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.
6. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x01, using a payload length of 10 with the payload octets set to 0x00. Repeat until all data sent in step 5 have been reported.
7. Upper Tester receives HCI_LE_Data_Packet events from the IUT containing the data element sent in step 6 with the Packet_Boundary_Flag flag set to 0x01.
8. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 in the first packet and 0x01 in the following. Using a payload length of 10 with the payload octets set to 0x00.
9. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02 in the first packet with data and 0x01 in the following, and using a payload length of 10 with the payload octets set to 0x00. Repeat until all data sent in step 8 has been reported.
10. Upper Tester receives an HCI_LE_Data_Packet event from the IUT with Packet_Boundary_Flag flag set to 0x02 and additional HCI_LE_Data_Packets events with Packet_Boundary_Flag flag set to 0x00. Payload octets shall be 0x00.

• Expected Outcome
  
  Pass Verdict
  
The test procedure completes with the IUT acknowledging all the data sent,

  The IUT reports all data correctly with HCI_Data_Packet events using the HCI fragmentation flags as specified in Section 4.1.6.

4.3.4.4 LL/CON/SLA/BV-06-C [Slave Sending and Receiving Data]

• Test Purpose
  Tests that a slave IUT is able to send and receive data to/from a master device.
  
The Lower Tester acts in the master role in the connection, both submits data for the IUT to transmit and sends data to it according to the acknowledgement scheme and observes the data received and reported to the host of the IUT.

• Reference
  
  [3] 4.5

• Initial Condition
  
  Parameters: LL_slave_payload_length_MIN, LL_slave_payload_length_MAX
  
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

• Test Procedure
  
  Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps. The number of octets transferred is over the data packet length of the IUT, with a minimum of 1000 octets.
1. Upper Tester submits data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set to 0x00 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.

2. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.

3. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00.

4. Lower Tester receives a DATA packet in response from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1.

5. Repeat steps 3–4 until all data sent in step 1 and 2 have been received/reported.

6. Upper Tester receives an HCI_Data_Packet event from the IUT containing a data element sent in step 1: with the Packet_Boundary_Flag flag set to 0x02 until all data sent in 1 have been reported.

- **Expected Outcome**

**Pass Verdict**

The test procedure completes with the IUT acknowledging all the data sent and reporting all data received.

The IUT sends the data preserving the fragmentation flags in the LLID field, as specified in Section 4.1.6.

The IUT reports the data preserving the upper layer messages in HCI_Data_Packet events.

### 4.3.4.5 LL/CON/SLA/BV-10-C [Accepting Parameter Update]

- **Test Purpose**

Tests that a slave IUT accepts a connection parameter update packet from a master device and starts using the new parameters in the event requested.

The Lower Tester acts in the master role in the connection, sending a connection parameter update packet to the IUT until it accepts it, then takes the new parameters into use. The Lower Tester observes the slave responding in all events until the update.
• Reference

[3] 5.1.1

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

• Test Procedure

Execute the test procedure using the common data channel selection parameters. Make the event count value minimum when changing from a short interval and maximum when changing from a long interval:

An update from the common connection interval values to the intermediate values using an event count of 100,

An update from intermediate connection values (100 ms interval, 2 slave latency, 3 s connection supervision timeout) back to the common connection interval using an event count of 10.
1. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the minimum connection interval, no latency, intermediate connection supervision timeout, and event count value maximum values. Lower Tester receives a packet from the IUT acknowledging the connection update request.

2. Lower Tester sends empty DATA packets to the IUT, receiving responses until the event count matches the indicated time of connection update:

3. At the time of the update start maintaining the connection with the new parameters.
4. Interleave with step 3: Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.

5. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum connection interval, no latency, intermediate connection supervision timeout, and event count value minimum values. Expect a packet from the IUT acknowledging the connection update request.

6. Repeat steps 2–4

7. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the minimum connection interval, no latency, intermediate connection supervision timeout, and event count value maximum values. Expect a packet from the IUT acknowledging the connection update request.

8. Repeat steps 2–4.

• Test Condition
  The parameters in this test are calculated for a BER of 0.1 percent or better.

• Expected Outcome
  Pass Verdict
  The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event,
  The IUT responds in at least 65 of the 100 events before the assigned event, when observed in the connection update from the short connection interval to the long interval,
  The IUT reports the new connection parameters with an HCI event.

• Notes
  The response rate (the specific percentage 88.7 %) from the IUT is assuming a BER of 0.1 % on the preamble and access address from the Lower Tester and the empty data packet from the IUT, for a length of 120 bits, with a near 100% confidence level in the test procedure gives around 65 out of 100. This accuracy is rough, it may be improved by:
  - Increasing the number of events (the packet count) in the update, but this results in over 1 s testing time,
  - Decreasing the reliability of the test case, making fewer repetitions under in conditions of interference.
  - Neither of the above is desirable, therefore as the 80 % is more than the slave latency it is a satisfactory result.

4.3.4.6  LL/CON/SLA/BV-11-C [Slave Sending Termination]

• Test Purpose
  Tests that a slave IUT is able to terminate a connection by sending the termination packet.
  The Lower Tester acts in the master role in the connection, receiving the termination packet from the IUT. The Lower Tester observes that there are no more packets from the slave once the termination packet is acknowledged.

• Reference
  [3] 5.1.6
• Initial Condition
State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, 100 ms timeout, common channel map, any SCA value)

• Test Procedure
Execute the test procedure using 0 slave latency and a 100 ms timeout.

Figure 4.170: LL/CON/SLA/BV-11-C [Slave sending termination]

1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps’ execution and receives an HCI_Command_Status in response.
2. Lower Tester expects the IUT to respond to a master transmission with a TERMINATE_IND packet. Lower Tester acknowledges the termination packet with an empty DATA packet in the following event.
3. Lower Tester sends empty DATA packets up to a time equal to the connection supervision timeout and receives no response to the master transmissions:
4. Interleave with step 3: Upper Tester receives an HCI_Disconnection_Complete event from the IUT indicating that the connection termination procedure requested in step 1 was successful.

• Expected Outcome
Pass Verdict
The test procedure executes successfully, with the IUT sending termination packets (success error code) until acknowledgement from the Lower Tester.

The IUT stops maintaining the connection once the TERMINATE_IND packet has been acknowledged by the Lower Tester.

The IUT reports the connection termination with an HCI event.

The IUT stops sending TERMINATE_IND packets when T_Terminate timer expires.
4.3.4.7 LL/CON/SLA/BV-12-C [Slave Accepting Termination]

- **Test Purpose**
  Test that a slave IUT accepts the termination from the master transmissions.
  The Lower Tester acts in the master role in the connection, sending the termination packets until receiving acknowledgement from the IUT.

- **Reference**
  [3] 5.1.6

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- **Test Procedure**
  Execute the test procedure using 0 slave latency and a 100 ms timeout.

  ![Diagram of LL/CON/SLA/BV-12-C Test Procedure]

  Figure 4.171: LL/CON/SLA/BV-12-C [Slave accepting termination]

  1. Lower Tester sends a TERMINATE_IND packet to the IUT and receives an empty DATA packet in response acknowledging the TERMINATE_IND packet.
  2. Lower Tester sends empty DATA packets up to a time equal to the connection supervision timeout and receives no response to the master transmissions.
  3. Interleave with step 2: Upper Tester receives an HCI_Disconnection_Complete event from the IUT indicating termination requested by the peer device and containing the connection handle from the preamble.

- **Expected Outcome**
  **Pass Verdict**
  The IUT acknowledges one termination packet, before T_Terminate timer expires.
  The IUT stops maintaining the connection once it has acknowledged the TERMINATE_IND packet.
  The IUT reports the connection termination with an HCI event.
4.3.4.8 LL/CON/SLA/BV-13-C [Slave Supervision Timer]

- **Test Purpose**
  Tests that a slave IUT terminates the connection from the connection supervision timer.
  The Lower Tester acts in the master role in the connection, stops transmitting events and observes the IUT notifying that the connection has been terminated.

- **Reference**
  [3] 4.5.2

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 slave latency, selected timeout, common channel map, any SCA value)

- **Test Procedure**

  ![Diagram of LL/CON/SLA/BV-13-C](image)

  **Figure 4.172: LL/CON/SLA/BV-13-C [Slave supervision timer]**

  1. The connection is established using the maximum connection supervision timer and connection interval values supported.
  2. Lower Tester stops maintaining the connection as a master after a specific event.
  3. Upper Tester receives an HCI_Disconnection_Complete_event from the IUT, indicating a connection supervision timeout and containing the connection handle used, after the last connection event before the connection supervision timeout expires (time equal to the connection supervision timeout selected from the last event transmitted).
  4. The connection is established using the minimum connection supervision timer and connection interval values supported.
  5. Repeat steps 2–3.

- **Expected Outcome**

  **Pass Verdict**
  The IUT produces the connection termination HCI event after the selected timeout values for the connSupervisionTimeout from the event.
• Notes

The required accuracy for the connection supervision timer is milliseconds. The drift with 50 ppm to the longest setting of connection supervision timeout, milliseconds is affected. Jitter does not affect the accuracy required.

The deviation by drift is in the order of 1 ms, making that the measurement accuracy required. Measurement results are rounded to tens of milliseconds for the maximum connection supervision timeout value and to milliseconds to the minimum.

4.3.4.9 LL/CON/SLA/BV-14-C [Feature Setup Response]

• Test Purpose

Tests that a connected slave IUT performs the feature setup procedure, activating the correct features when requested.

The Lower Tester acts in the master role in a maintained connection, transmits the request to perform feature setup and observes the IUT responding.

• Reference

[3] 5.1.4

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, selected slave latency, selected timeout, common channel map, any SCA value).

• Test Procedure

1. Lower Tester sends an LL_FEATURE_REQ PDU including the Lower Tester's feature set and waits for an LL_FEATURE_RSP PDU.
2. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble step execution).
• Expected Outcome
  Pass Verdict
  All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.
  The test procedure is executed successfully, with the IUT responding with the feature information response.

4.3.4.10   LL/CON/SLA/BV-15-C [Slave Retransmission Request]

• Test Purpose
  Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme while receiving invalid checksums in data packets.
  The Lower Tester acts in the master role, starting events using the default connection parameters, using invalid checksums to prompt a repeated retransmission request from the IUT.

• Reference
  [3] 4.5.9

• Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

• Test Procedure
  Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

![Diagram](image)

*Figure 4.174: LL/CON/SLA/BV-15-C [Slave retransmission request]*

1. Lower Tester sends a DATA packet with invalid checksum once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet received is the next
SN and by using the next NESN where the current NESN matches the SN in the previous packet correctly received.

2. In events where the IUT is required to listen, Lower Tester receives a DATA packet T_IFS after the packet sent. Lower Tester expects the IUT to indicate in packets received correctly that the previous packet was not received correctly by the SN matching the current NESN and the NESN matching the current SN.

3. Repeat steps 1–2 15 times.

4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- Expected Outcome

  **Pass Verdict**

  The IUT transmits negative acknowledgements when sent a packet with an invalid CRC in 1.

### 4.3.4.11 LL/CON/SLA/BV-16-C [Slave Retransmission]

- **Test Purpose**

  Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and retransmit a data packet on a negative acknowledgement.

  The Lower Tester acts in the master role starting events using the default connection parameters and using negative acknowledgements prompts repeated retransmission requests from the IUT.

- **Reference**

  [3] 4.5.9

- **Initial Condition**

  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- **Test Procedure**

  Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.
1. Lower Tester sends an Empty DATA packet once a connection interval to the IUT using the data channel selection parameters has been previously established. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet received is the next SN, but omit acknowledgements by using the current NESN.

2. In events where the IUT is required to listen, the Lower Tester receives a DATA packet T_IFS after the packet sent. Expect packets received correctly from the IUT to retransmit the packet by the SN matching the current NESN and the NESN matching the next SN.

3. Repeat steps 1–2 15 times.

4. Lower Tester terminates the connection.

- Expected Outcome

**Pass Verdict**

The IUT retransmits the same SN and payload when asked for a retransmission.

### 4.3.4.12 LL/CON/SLA/BV-17-C [Slave Acknowledgement Repetition]

- **Test Purpose**

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and repeats a positive acknowledgement of a packet.

The Lower Tester acts in the master role, starting events using the default connection parameters, using negative acknowledgements to prompt a retransmission of an acknowledgement from the IUT.

- **Reference**

[3] 4.5.9

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value).
• **Test Procedure**

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

![Diagram](image_url)

*Figure 4.176: LL/CON/SLA/BV-17-C [Slave acknowledgement repetition]*

1. Lower Tester sends a DATA packet (with SN\_TESTER) once a connection interval to the IUT using the data channel selection parameters. Not to recognize an acknowledgement, use the current SN where in the previous packet received the NESN is the next SN, but acknowledge packets correctly by using the next NESN where the current NESN matches the SN in the previous packet correctly received.
2. The IUT sends an acknowledgement (with SN\_IUT and NESN\_IUT=NEXT SN\_TESTER).
3. Lower Tester does not recognize the acknowledgement and resend the DATA packet using the SN\_TESTER and payload equal to those of the previous packet sent by the Lower Tester.
4. Lower Tester expects the IUT to repeat the acknowledgement (SN\_IUT=NEXT SN\_IUT, NESN\_IUT=NEXT SN\_TESTER).
5. Repeat steps 3–4 15 times.
6. Lower Tester terminates the connection.

• **Expected Outcome**

**Pass Verdict**

The IUT retransmits an acknowledgement when being retransmitted the same payload.

4.3.4.13 **LL/CON/SLA/BV-18-C [Slave Lost Negative Acknowledgement]**

• **Test Purpose**

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and preserve the packet sequence numbering in the case of a lost negative acknowledgement.

The Lower Tester acts in the master role, starting events using the default connection parameters, moving to the next packet after a negative acknowledgement to prompt a repeated negative acknowledgement from the IUT.
• **Reference**

[3] 4.5.9

• **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

• **Test Procedure**

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

![Diagram of LL/CON/SLA/BV-18-C](image)

**Figure 4.177: LL/CON/SLA/BV-18-C [Slave lost negative acknowledgement]**

1. Lower Tester sends a DATA packet with a varying checksum once a connection interval to the IUT using the data channel selection parameters. To prompt the IUT to produce a negative acknowledgement, use the next SN for one packet to send with an invalid checksum, where the NESN in the previous packet received is the next SN, then if the following packet’s NESN is the current SN use the next SN with a valid checksum. The Lower Tester responds with a negative acknowledgment to the IUT’s packet by using a NESN that matches the SN of the IUT’s packet.

2. In events where the IUT is required to listen, Lower Tester receives a DATA packet T_IFS after the packet sent. Expect the IUT to repeat the packet, with the SN matching the current NESN and the NESN matching the current SN.

3. Repeat steps 1–2 15 times.

4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

• **Expected Outcome**

**Pass Verdict**

The IUT retransmits a negative acknowledgement when being transmitted the next payload.
4.3.4.14 LL/CON/SLA/BV-19-C [Slave Request Version]

- **Test Purpose**
  Test that a connected slave IUT requests and performs the Version Exchange procedure.
  The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to perform version exchange.

- **Reference**
  [3] 5.1.5

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- **Test Procedure**

  ![Diagram](Image)

  **Figure 4.178: LL/CON/SLA/BV-19-C [Slave Request Version]**

  1. Once connection is established, if the IUT has sent the Lower Tester a Version_Ind packet containing IUT version information by itself, the Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
  2. If the IUT does not send the Lower Tester a Version_Ind packet containing IUT version information by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command and receives an HCI_Command_Status_Event.
  3. The IUT sends to the Lower Tester a Version_Ind packet containing IUT version information.
  4. The Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
  5. The IUT sends to the Upper Tester the event HCI_Read_Remote_Version_Information_Complete_Event with Tester version information.
• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT requesting the version information and acknowledging the reply.

If the procedure was initiated by the Upper Tester as described in step 2, the IUT reports the version requested completed with an HCI event.

4.3.4.15 LL/CON/SLA/BV-20-C [Slave Respond Version]

• Test Purpose

Test that a connected slave IUT responds to the request from the Lower Tester to perform the version exchange procedure.

The Lower Tester acts in the master role in a maintained connection and requests to perform version exchange.

• Reference

[3] 5.1.5

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

• Test Procedure

![Diagram]

Figure 4.179: LL/CON/SLA/BV-20-C [Slave Respond Version]

1. Configure the Lower Tester to send Version_Ind packet
2. Lower Tester sends a Version_Ind packet containing Tester version information.
3. IUT acknowledges the Version_Ind packet and sends a Version_Ind packet containing IUT version information to the Lower Tester.

• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT responding to the version information.

Inconclusive Verdict

The IUT sends an LL_VERSION_IND PDU before the Lower Tester does.
4.3.4.16 LL/CON/SLA/BV-21-C [Slave Acknowledgement Scheme]

- **Test Purpose**
  Test that a slave IUT is able to maintain a connection observing the acknowledgement scheme.
  The Lower Tester acts in the master role, maintaining the connection and checking that the IUT uses correctly the acknowledgement scheme.

- **Reference**
  [3] 4.5.9

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- **Test Procedure**
  Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

1. Lower Tester sends a DATA packet once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received matches the next SN and by using the next NESN where the SN in the previous packet correctly received matches the current NESN.
2. In events where the IUT is required to listen, Lower Tester receives a DATA packet T_IFS after the packet sent, with the SN matching the current NESN and the NESN matching the next SN.
3. Repeat steps 1–2 15 times.
4. State: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).
• Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT using the normal acknowledgment scheme operation.

The IUT ACKs at least to 10 out of 15 packets sent by the Lower Tester.

4.3.4.17 LL/CON/SLA/BV-22-C [Initiate Feature Exchange]

• Test Purpose

Test that a connected slave IUT requests and performs the feature exchange procedure, activating the correct features.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to perform feature exchange.

• Reference

[3] 5.1.4.2

• Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

• Test Procedure

![Diagram](attachment://Figure_4.181_LL%2FCON%2FSLA%2FBV-22-C_Initiate_Feature_Exchange.png)
1. Upper Tester sends an HCI_LE_Read_Local_Supported_Features command and receives an HCI_Command_Complete event with the correct feature set value.

2. Upper Tester sends an HCI_LE_Read_Remote_Features command and receives an HCI_Command_Status event.

3. The IUT may have autonomously sent an LL_SLAVE_FEATURE_REQ prior to receiving the HCI command from the Upper Tester; otherwise, the IUT sends an LL_SLAVE_FEATURE_REQ PDU, including the configured feature set and waits for an LL_FEATURE_RSP PDU from the Lower Tester.

4. The IUT sends the HCI_LE_Read_Remote_Features_Complete event to the Upper Tester.

**Expected Outcome**

**Pass Verdict**

All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT requesting the feature information and acknowledging the reply.

The IUT reports the feature exchange procedure completed with an HCI_LE_Read_Remote_Features_Complete event.

### 4.3.4.18 LL/CON/SLA/BV-23-C [Initiate Feature Exchange – Master does not support]

**Test Purpose**

Test that a slave IUT requests and performs the feature exchange procedure when the master does not support that procedure.

The Lower Tester acts in the master role in the connection and responds to the request from the IUT with an LLUNKNOWN_RSP.

**Reference**

[3] 5.1.4.2

**Initial Condition**

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

1. Upper Tester sends an HCI_LE_Read_Remote_Feature command and receives an HCI_Command_Status_Event.
2. The IUT may have autonomously sent an LL_SLAVE_FEATURE_REQ prior to receiving the HCI command from the Upper Tester; otherwise, the IUT sends an LL_SLAVE_FEATURE_REQ PDU, including the configured feature set and waits for an LL_FEATURE_RSP PDU from the Lower Tester.
3. Lower Tester acknowledges the LL_SLAVE_FEATURE_REQ PDU and sends an LL_UNKNOWN_RSP PDU in response.
4. The IUT sends the HCI_LE_Read_Remote_Feature_Complete event to the Upper Tester with the correct error code.

**Note**

The Lower Tester must not initiate a feature exchange during the test procedure after the connection has been established.

**Expected Outcome**

**Pass Verdict**

The IUT sends HCI_Read_Remote_Feature_Complete (Reason: 0x1A) when the Lower Tester responds with an LL_UNKNOWN_RSP PDU.

4.3.4.19 **LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept]**

• Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device accepts the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and accepts the IUT’s request. The actual parameters used by the IUT may be different from the parameters provided by the Upper Tester.
• Reference

[3] 5.1.7

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own
address, any advertising channel map, connection interval greater than
LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any
SCA value).
• Test Procedure

Connection Established. IUT Slave.

Figure 4.183: LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept]
Case 1:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the maximum connection interval, no latency and maximum connection supervision timeout and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

For all three cases described in the test procedure all the following three conditions shall occur:

- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and uses the new parameters selected by the IUT at the assigned event.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.20 LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject]

• Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device rejects the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT’s request.

• Reference

[3] 5.1.7

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

• Test Procedure

![Diagram of test procedure](attachment:image)

*Figure 4.184: LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject]*
1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.

2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester) and the Lower Tester rejects the IUT’s request by issuing an LL_REJECT_EXT_IND PDU. Lower Tester receives a packet from the IUT acknowledging the LL_REJECT_EXT_IND PDU.

3. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the error code sent by the Lower Tester.

- **Expected Outcome**

  **Pass Verdict**

  The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.

  The IUT reports an HCI_LE_Connection_Update_Complete event containing the error code sent by the Lower Tester.

  **4.3.4.21 LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision]**

- **Test Purpose**

  Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT’s connection parameter request and the Lower Tester’s connection parameter request.

  The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a new connection parameter request procedure upon receiving the IUT’s connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master’s connection parameter request procedure after the master has rejected the IUT’s connection parameter request procedure.

- **Reference**

  [3] 5.1.7

- **Initial Condition**

  Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
- **Test Procedure**

![Diagram](image)

**Figure 4.185: LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision]**

1. **Upper Tester** sends an HCI_LE_Connection_Update command to the **IUT** setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.

2. **Lower Tester** receives an LL_CONNECTION_PARAM_REQ control PDU from the **IUT** (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). **Lower Tester** responds with an LL_CONNECTION_PARAM_RSP and rejects the **IUT**'s LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x23.

3. The **IUT** requests the **Upper Tester** to accept or reject the **Lower Tester**'s request.

4. At this point or later in the procedure, the **IUT** may optionally send an HCI_LE_Connection_Update_Complete_event with reason code set to 0x23 to the **Upper Tester**.

5. The **Upper Tester** accepts the **IUT**'s request to accept or reject the **Lower Tester**'s request.
6. The IUT responds to the LL_CONNECTION_PARAM_REQ from the Lower Tester with an LL_CONNECTION_PARAM_RSP.
7. The Lower Tester responds with an LL_CONNECTION_UPDATE_IND PDU and receives a packet from the IUT acknowledging the connection update request.
8. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
9. At the time of the update start maintaining the connection with the new parameters.
10. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

  **Pass Verdict**
  - The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
  - If the IUT reports an HCI_LE_Connection_Update_Complete event when the Lower Tester rejects the IUT’s request, it shall have the correct error code (0x23).
  - The IUT responds positively to the master initiated connection parameter request procedure.
  - The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event.
  - The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.22 LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

- Test Purpose

  Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT's connection parameter request and the remote device's channel map update.

  The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a channel map update procedure upon receiving the IUT’s connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master’s channel map update procedure after the master has rejected the IUT’s connection parameter request procedure.

- Reference

  [3] 5.1.7

- Initial Conditions

  Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

![Diagram of connection establishment process]

**Figure 4.186: LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]**

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_CHANNEL_MAP_IND and rejects the IUT's LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND.
3. IUT sends the HCI_LE_Connection_Update_Complete event with reason code set to 0x2A to the Upper Tester.
4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of channel map update.

• Expected Outcome

**Pass Verdict**

- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and reports an HCI_LE_Connection_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT's request.
- The IUT responds positively to the master initiated channel map request procedure.
- The test procedure is executed successfully, with the IUT acknowledging the channel map request and adopting the new channel map at the assigned event.
- The IUT maintains the connection with the Lower Tester after the channel map update completes.

4.3.4.23 LL/CON/SLA/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT’s connection parameter request and the remote device’s encryption procedure.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates an encryption start procedure upon receiving the IUT’s connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master’s encryption start procedure and then complete the IUT’s connection parameter request procedure.

- Reference

[3] 5.1.7, 5.1.3.1

- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
Test Procedure

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_ENC_REQ PDU.
3. The Lower Tester expects the IUT to respond with an LL_ENC_RSP.
4. The Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT.
5. The Upper Tester sends and HCI_LE_Long_Term_Key_Requested_Reply to the IUT and receives an HCI_Command_Complete in response.
6. The Lower Tester receives an LL_START_ENC_REQ from the IUT which it acknowledges.
7. The Lower Tester sends an LL_START_ENC_RSP PDU to the IUT and receives an LL_START_ENC_RSP PDU from the IUT in response.
8. The Upper Tester receives an HCI_Encryption_CHANGE event with encryption enable set to on.
9. The Lower Tester responds to the IUT’s connection parameter request with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
10. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
11. At the time of the update start maintaining the connection with the new parameters.
12. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict
- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
- The IUT sends its initialization vector and session key diversifier in a LL_ENC_RSP packet.
- The IUT reports the encryption setup requested with the HCI event HCI_LE_Long_Term_Key_Requested.
- The IUT sends a LL_START_ENC_REQ packet until acknowledged.
- The IUT acknowledges the LL_START_ENC_RSP and responds with one.
- The IUT successfully reports the encryption change with the HCI event HCI_Encryption_Change.
- The IUT acknowledges the connection update request from the Lower Tester and adopts the new parameters selected by the IUT at the assigned event.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

- Notes

The Lower Tester and Upper Tester ensure that the encryption start procedure completes before the procedure response timeout for the connection parameter request procedure fires.

4.3.4.24 LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master does not indicate any preferred periodicity.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

![Test Procedure Diagram]

Figure 4.188: LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity]
Case 1:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and PreferredPeriodicity set to zero.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTIONPARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTIONPARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTIONUPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the maximum connection interval, no latency, and maximum connection supervision timeout and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTIONPARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTIONPARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTIONUPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTIONPARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTIONPARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTIONUPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

For all three cases described in the test procedure, the following conditions shall occur:

- The IUT responds positively to the Lower Tester’s request to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.25 LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only]

• Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master only requests a change in anchor points.

• Reference

[3] 5.1.7

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

![Diagram of LL/CON/SLA/BV-30-C test procedure]

**Figure 4.189: LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only]**

**Case 1:**

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to invalid and the connection interval, latency and supervision timeout unchanged.

2. Lower Tester receives an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are 1.25ms from the old anchor points. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

Case 2:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to (connection interval – 1.25ms), Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester receives an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are (connection interval - 1.25ms) from the old anchor points. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

Case 3:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to 2.5ms, Offset2-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester receives an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are 1.25ms from the old anchor points. Lower Tester receives a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

- Expected Outcome

Pass Verdict
- The IUT responds positively to the Lower Tester’s request to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event:
  - In the first case, the anchor points are shifted by 1.25ms.
  - In the second case, the anchor points are shifted by (connection interval - 1.25ms).
  - In the third case, the anchor points are shifted by 1.25ms.

4.3.4.26 LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity]

- Test Purpose
  Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master indicates a preferred periodicity.

- Reference
  [3] 5.1.7
Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

Test Procedure

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s) and a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update start maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Figure 4.190: LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity]
• Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester’s request to update connection parameters.

The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.27  LL/CON/SLA/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points]

• Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master indicates a preferred periodicity and preferred anchor points.

• Reference

[3] 5.1.7

• Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s), a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range, a reference connection event counter and a valid Offset0 value such that the new connection event is 1.25 ms away from the old connection event at the reference connection event count (Offset1-5 are invalid).

2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.

3. Lower Tester receives an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester receives a packet from the IUT acknowledging the connection update request.

4. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.

5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.

6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

• Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester's request to update connection parameters.
The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.28 LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked]

- **Test Purpose**

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master requires the slave LL to request for approval from the slave’s Host and the slave’s Host has masked the LE Remote Connection Parameter Request Event.

- **Reference**

[3] 5.1.7

- **Initial Condition**

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- **Test Procedure**

1. Upper Tester masks the LE Remote Connection Parameter Request event on the IUT.
2. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
3. Lower Tester receives an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1A.

---

*Figure 4.192: LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked]*

- **Lower Tester**
- **IUT**
- **Upper Tester**
• Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester’s request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode (0x1A).

4.3.4.29 LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects]

• Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the slave’s Host rejects the master’s connection parameter request procedure.

• Reference

[3] 5.1.7

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

• Test Procedure

1. Lower Tester sends an LL_CONNECTION PARAM REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester rejects the request using ErrorCode 0x3B.
3. Lower Tester receives an LL_REJECT_EXT_IND control PDU from the IUT containing the ErrorCode provided by the Upper Tester.

Figure 4.193: LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects]
• Expected Outcome
  
  Pass Verdict
  
  The IUT responds to the Lower Tester’s request to update connection parameters with an LL_REJECT_EXT_IND using the ErrorCode provided by the Upper Tester.

4.3.4.30  LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure]

• Test Purpose
  
  Test that a slave IUT is able to perform the PHY update procedure. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

  The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and accepts the IUT’s request.

• Reference
  
  [10] 5.1.10

• Initial Condition
  
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

• Test Procedure

  Figure 4.194: LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure]
The following steps shall be carried out 2N times as follows, where N is the number of cases in Table 4.52:

- firstly using cases 1 to N from Table 4.52 in order;
- then using the cases from Table 4.52 in a random order.

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the payload defined in the HCI_LE_Set_PHY section of Table 4.52 and PHY_options set to 0x0000.
2. The Upper Tester receives an HCI_Command_Status event from the IUT in response. If any bits set in TX_PHYS or RX_PHYS correspond to unsupported PHYs, the Status shall be set to “Unsupported Feature or Parameter Value (0x11)”. If the IUT does not support Asymmetric Connections, when ALL_PHYS is 0x00 and TX_PHYS does not equal RX_PHYS, the Status shall be set to “Unsupported Feature or Parameter Value (0x11)”. Otherwise the Status shall be set to zero.
3. If the IUT does NOT initiate a PHY change, proceed to step 9 if the Status in step 2 was set to zero, or proceed to the next round if the Status in step 2 was set to a non-zero value.
4. The Lower Tester receives an LL_PHY_REQ control PDU from the IUT with at least one bit in each field (TX_PHYS, RX_PHYS) set. The Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the values defined in the “Lower Tester preference” section of Table 4.52 bitwise ANDed against the value sent by the IUT in the LL_PHY_REQ PDU such that:

- M_TO_S_PHY = (M_TO_S_PHY_LTPREF & (LL_PHY_REQ RX_PHYS Field))
- S_TO_M_PHY = (S_TO_M_PHY_LTPREF & (LL_PHY_REQ TX_PHYS Field))
- If the IUT specifies the same single PHY in both the RX_PHYS and TX_PHYS fields, the Lower Tester shall use the PHY selected by the IUT for both directions or make no change. This rule shall take precedence over the values determined in rules (a) and (b).
- If any of rules (a) to (c) result in a PHY equal to the current PHY, the corresponding field shall be set to zero rather than the bit corresponding to that PHY.
- If either M_TO_S_PHY or S_TO_M_PHY are non-zero, then (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If M_TO_PHY and S_TO_M_PHY are both zero, the Instant shall be zero.
5. Lower Tester receives a packet from the IUT acknowledging the LL_PHY_UPDATE_IND.
6. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
7. At the Instant of the PHY change, start maintaining the connection with the new PHY(s) selected by the LL_PHY_UPDATE_IND PDU (or no change, if no change was specified in the LL_PHY_UPDATE_IND PDU).
8. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements. If PHY has changed, the Lower Tester shall use the new PHY.
9. If the command was accepted in step 2 or at least one of the PHY fields in the LL_PHY_UPDATE_IND PDU was non-zero, the Upper Tester receives an LE_PHY_Update_Complete event from the IUT with a payload consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in the LL_PHY_UPDATE_IND PDU was zero or the LL_PHY_UPDATE_IND PDU was not sent). Otherwise the Upper Tester receives no event.

<table>
<thead>
<tr>
<th>Case</th>
<th>HCI_LE_Set_PHY</th>
<th>Lower Tester preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
<td>RX_PHYS</td>
</tr>
<tr>
<td>1</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>Case</td>
<td>HCI_LE_Set_PHY</td>
<td>Lower Tester preference</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>2</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>3</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>4</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>5</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>6</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>7</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>8</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>9</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>10</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>11</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>12</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>13</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>14</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>15</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>16</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>17</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>18</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>19</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>20</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>21</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>22</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>23</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>24</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>25</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>Case</td>
<td>HCI_LE_Set_PHY</td>
<td>Lower Tester preference</td>
</tr>
<tr>
<td>------</td>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>26</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>27</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>28</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>29</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>30</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>31</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>32</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>33</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>34</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>35</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>36</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>37</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>38</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>39</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>40</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>41</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>42</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>43</td>
<td>0x00</td>
<td>0x07</td>
</tr>
<tr>
<td>44</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>45</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>46</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>47</td>
<td>0x02</td>
<td>0x03</td>
</tr>
<tr>
<td>48</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>49</td>
<td>0x02</td>
<td>0x05</td>
</tr>
</tbody>
</table>
### Table 4.52: PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported.

<table>
<thead>
<tr>
<th>Case</th>
<th>HCI_LE_Set_PHY</th>
<th>Lower Tester preference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>50</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>51</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>52</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>53</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>54</td>
<td>0x02</td>
<td>0x07</td>
</tr>
<tr>
<td>55</td>
<td>0x02</td>
<td>0x06</td>
</tr>
<tr>
<td>56</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>57</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>58</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>59</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>60</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>61</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>62</td>
<td>0x00</td>
<td>0x00</td>
</tr>
<tr>
<td>63</td>
<td>0x02</td>
<td>0x11</td>
</tr>
<tr>
<td>64</td>
<td>0x02</td>
<td>0x20</td>
</tr>
<tr>
<td>65</td>
<td>0x02</td>
<td>0x41</td>
</tr>
<tr>
<td>66</td>
<td>0x02</td>
<td>0x80</td>
</tr>
<tr>
<td>67</td>
<td>0x02</td>
<td>0xFF</td>
</tr>
</tbody>
</table>

- **Expected Outcome**

  **Pass Verdict**

  For all cases described in the test procedure all the following conditions shall occur:
  - If the IUT transmits an LL_PHY_REQ PDU, at least one bit shall be set in each field (TX_PHYS, RX_PHYS).
  - If the IUT is symmetric only, then it shall always select a single (symmetric) PHY in each LL_PHY_REQ PDU.
- If IUT transmits an LL_PHY_REQ, the change procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and using the PHY(s) selected by the LL_PHY_UPDATE_IND at the assigned event and thereafter.

- The IUT reports the currently selected PHY with a LE_PHY_Update_Complete event with the RX_PHY and TX_PHY fields consistent with the PHY(s) selected via the LL_PHY_UPDATE_IND PDU if either the command was accepted in step 2 or a PHY change occurred. If no change occurs, the RX_PHY and TX_PHY fields shall reflect the previous PHY still in use.

- The IUT does not send a LE_PHY_Update_Complete event if the command was rejected in step 2 and either the IUT did not initiate the PHY Update Procedure or it initiated the procedure but no PHY change occurred.

Fail Verdict
The IUT accepts the command in step 2 when a bit set in TX_PHYS or RX_PHYS corresponds to an unsupported PHY.

Inconclusive Verdict
The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.4.31 LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure]

• Test Purpose
Test that a slave IUT is able to respond to a PHY update procedure from a master device. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the master role maintaining a connection and initiates the PHY update procedure. IUT responds to the PHY change request and notifies the host of the change only when appropriate.

• Reference
[10] 5.1.10

• Initial Condition
State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• **Test Procedure**

![Diagram of test procedure]

**Figure 4.195: LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure]**

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the ALL_PHYS fields set to a value of 0x03. Upper Tester receives an HCI_Command_Status event indicating success in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT.
3. Perform steps 4 through 11 2N times as follows, where N is the number of cases in Table 4.53, Table 4.54, or Table 4.55 (selected based on the supported PHY(s)):
   - Firstly using cases 1 to N from the relevant table in order;
   - Then using the cases from the relevant table in a random order.
4. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY change with the payload defined in the LL_PHY_REQ section of the relevant table.
5. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS).
6. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the payload defined by the following rules:
   a. \( M_{TO\_S\_PHY} = (LL\_PHY\_REQ\ TX\_PHY) \& (LL\_PHY\_RSP\ RX\_PHY) \). If this value has 0 or 1 bit set, use the value. If more than one bit is set, the Lower Tester shall select the bit listed in the M_TO_S_PHY_LTPREF column of the Lower Tester preference section of the
relevant table; if this table lists more than one bit, it shall select the first bit (in the order given) that is set.

b. \( S_{TO\_M\_PHY} = (LL\_PHY\_REQ\ RX\_PHYS) \& (LL\_PHY\_RSP\ TX\_PHYS\ Field) \). If this value has 0 or 1 bit set, use the value. If more than one bit is set, the Lower Tester shall select the bit listed in the \( S_{TO\_M\_PHY\_LTPREF} \) column of the Lower Tester preference section of the relevant table; if this table lists more than one bit, it shall select the first bit (in the order given) that is set.

c. If IUT specifies the same single PHY in both the \( RX\_PHYS \) and \( TX\_PHYS \) fields, the Lower Tester shall use the PHY selected by the IUT for both directions (if allowed by the \( LL\_PHY\_REQ \)) or make no change. This rule shall take precedence over the values determined in rules (a) and (b).

d. If any of rules (a) to (c) result in a PHY equal to the current PHY, the corresponding field shall be set to zero rather than the bit corresponding to that PHY.

e. If either \( M\_TO\_S\_PHY \) or \( S\_TO\_M\_PHY \) are non-zero, then \((\text{Instant} – \text{connEventCount}) \mod 65536\) shall be less than 32767 and greater than 6. If \( M\_TO\_PHY \) and \( S\_TO\_M\_PHY \) are both zero, the Instant shall be zero.

7. Lower Tester receives a packet from the IUT acknowledging the \( LL\_PHY\_UPDATE\_IND \). If both the \( M\_TO\_S\_PHY \) and \( S\_TO\_M\_PHY \) fields of the \( LL\_PHY\_UPDATE\_IND \) are zero, skip to step 11.

8. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated Instant of the PHY change.

9. At the Instant of the PHY change the IUT starts maintaining the connection with the new PHY(s) selected by the Lower Tester.

10. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements. If the PHY(s) have changed, the Lower Tester shall use the new PHY(s).

11. If the PHY(s) were changed, Upper Tester receives a \( LE\_PHY\_Update\_Complete \) event from the IUT containing the PHYs selected. If both PHYs were NOT changed, Upper Tester does NOT receive a \( LE\_PHY\_Update\_Complete \) event.

<table>
<thead>
<tr>
<th>Case</th>
<th>LL_PHY_REQ</th>
<th>Lower Tester preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX_PHYS</td>
<td>RX_PHYS</td>
</tr>
<tr>
<td>1</td>
<td>0x02</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x01</td>
<td>0x02</td>
</tr>
<tr>
<td>3</td>
<td>0x02</td>
<td>0x01</td>
</tr>
<tr>
<td>4</td>
<td>0x01</td>
<td>0x01</td>
</tr>
<tr>
<td>5</td>
<td>0x03</td>
<td>0x02</td>
</tr>
<tr>
<td>6</td>
<td>0x03</td>
<td>0x01</td>
</tr>
<tr>
<td>7</td>
<td>0x01</td>
<td>0x03</td>
</tr>
</tbody>
</table>
### Table 4.53: PDU payload contents for each case variation for LE 2M PHY supported and LE Coded PHY not supported.

<table>
<thead>
<tr>
<th>Case</th>
<th>LL_PHY_REQ</th>
<th>Lower Tester preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX_PHYS</td>
<td>RX_PHYS</td>
</tr>
<tr>
<td>8</td>
<td>0x02</td>
<td>0x03</td>
</tr>
<tr>
<td>9</td>
<td>0x03</td>
<td>0x03</td>
</tr>
</tbody>
</table>

### Table 4.54: PDU payload contents for each case variation for LE Coded PHY supported and LE 2M PHY not supported.

<table>
<thead>
<tr>
<th>Case</th>
<th>LL_PHY_REQ</th>
<th>Lower Tester preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX_PHYS</td>
<td>RX_PHYS</td>
</tr>
<tr>
<td>1</td>
<td>0x04</td>
<td>0x04</td>
</tr>
<tr>
<td>2</td>
<td>0x01</td>
<td>0x04</td>
</tr>
<tr>
<td>3</td>
<td>0x04</td>
<td>0x01</td>
</tr>
<tr>
<td>4</td>
<td>0x01</td>
<td>0x01</td>
</tr>
<tr>
<td>5</td>
<td>0x05</td>
<td>0x04</td>
</tr>
<tr>
<td>6</td>
<td>0x05</td>
<td>0x01</td>
</tr>
<tr>
<td>7</td>
<td>0x01</td>
<td>0x05</td>
</tr>
<tr>
<td>8</td>
<td>0x04</td>
<td>0x05</td>
</tr>
<tr>
<td>9</td>
<td>0x05</td>
<td>0x05</td>
</tr>
<tr>
<td>Case</td>
<td>LL_PHY_REQ</td>
<td>Lower Tester preferences</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>TX_PHY</td>
<td>RX_PHY</td>
</tr>
<tr>
<td>7</td>
<td>0x01</td>
<td>0x01</td>
</tr>
<tr>
<td>8</td>
<td>0x04</td>
<td>0x03</td>
</tr>
<tr>
<td>9</td>
<td>0x05</td>
<td>0x01</td>
</tr>
<tr>
<td>10</td>
<td>0x04</td>
<td>0x04</td>
</tr>
<tr>
<td>11</td>
<td>0x05</td>
<td>0x07</td>
</tr>
<tr>
<td>12</td>
<td>0x05</td>
<td>0x05</td>
</tr>
<tr>
<td>13</td>
<td>0x04</td>
<td>0x02</td>
</tr>
<tr>
<td>14</td>
<td>0x03</td>
<td>0x07</td>
</tr>
<tr>
<td>15</td>
<td>0x06</td>
<td>0x06</td>
</tr>
<tr>
<td>16</td>
<td>0x03</td>
<td>0x02</td>
</tr>
<tr>
<td>17</td>
<td>0x01</td>
<td>0x06</td>
</tr>
<tr>
<td>18</td>
<td>0x05</td>
<td>0x06</td>
</tr>
<tr>
<td>19</td>
<td>0x04</td>
<td>0x05</td>
</tr>
<tr>
<td>20</td>
<td>0x01</td>
<td>0x05</td>
</tr>
<tr>
<td>21</td>
<td>0x05</td>
<td>0x03</td>
</tr>
<tr>
<td>22</td>
<td>0x01</td>
<td>0x04</td>
</tr>
<tr>
<td>23</td>
<td>0x01</td>
<td>0x03</td>
</tr>
<tr>
<td>24</td>
<td>0x03</td>
<td>0x05</td>
</tr>
<tr>
<td>25</td>
<td>0x06</td>
<td>0x04</td>
</tr>
<tr>
<td>26</td>
<td>0x02</td>
<td>0x07</td>
</tr>
<tr>
<td>27</td>
<td>0x06</td>
<td>0x01</td>
</tr>
<tr>
<td>28</td>
<td>0x02</td>
<td>0x02</td>
</tr>
<tr>
<td>29</td>
<td>0x03</td>
<td>0x04</td>
</tr>
<tr>
<td>30</td>
<td>0x07</td>
<td>0x03</td>
</tr>
<tr>
<td>Case</td>
<td>LL_PHY_REQ</td>
<td>Lower Tester preferences</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>TX_PHYS</td>
<td>RX_PHYS</td>
</tr>
<tr>
<td>31</td>
<td>0x02</td>
<td>0x01</td>
</tr>
<tr>
<td>32</td>
<td>0x03</td>
<td>0x03</td>
</tr>
<tr>
<td>33</td>
<td>0x02</td>
<td>0x03</td>
</tr>
<tr>
<td>34</td>
<td>0x04</td>
<td>0x07</td>
</tr>
<tr>
<td>35</td>
<td>0x07</td>
<td>0x04</td>
</tr>
<tr>
<td>36</td>
<td>0x07</td>
<td>0x01</td>
</tr>
<tr>
<td>37</td>
<td>0x06</td>
<td>0x05</td>
</tr>
<tr>
<td>38</td>
<td>0x02</td>
<td>0x06</td>
</tr>
<tr>
<td>39</td>
<td>0x07</td>
<td>0x07</td>
</tr>
<tr>
<td>40</td>
<td>0x04</td>
<td>0x06</td>
</tr>
<tr>
<td>41</td>
<td>0x02</td>
<td>0x05</td>
</tr>
<tr>
<td>42</td>
<td>0x06</td>
<td>0x02</td>
</tr>
<tr>
<td>43</td>
<td>0x07</td>
<td>0x02</td>
</tr>
<tr>
<td>44</td>
<td>0x07</td>
<td>0x06</td>
</tr>
<tr>
<td>45</td>
<td>0x06</td>
<td>0x07</td>
</tr>
<tr>
<td>46</td>
<td>0x06</td>
<td>0x03</td>
</tr>
<tr>
<td>47</td>
<td>0x05</td>
<td>0x04</td>
</tr>
<tr>
<td>48</td>
<td>0x07</td>
<td>0x05</td>
</tr>
<tr>
<td>49</td>
<td>0x01</td>
<td>0x07</td>
</tr>
</tbody>
</table>

Table 4.55: PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported.

- **Expected Outcome**

  **Pass Verdict**

  For all cases described in the test procedure, the following conditions shall occur:

  - The IUT responds to the Lower Tester’s LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHY, RX_PHY).
The test procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and adopting the new PHY(s) selected by the Lower Tester at the assigned event if the PHY(s) are changed, or using the prior PHY(s) if the PHY(s) are NOT changed.

- If the PHY(s) are changed, the IUT reports the selected PHY with a LE_PHY_Update_Complete event. If both PHYs are NOT changed, the IUT does NOT send a LE_PHY_Update_Complete event. If a LE_PHY_Update_Complete event is sent, its fields are consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in LL_PHY_UPDATE_IND was zero).

**Inconclusive Verdict**

The PHY does not change (equivalently, step 9 is not carried out) at least once during this test case because of the rules in step 6.

### 4.3.4.32 LL/CON/SLA/BV-43-C [Responding to PHY Update Procedure – Symmetric Only]

**Test Purpose**

Test that a slave IUT is able to respond to a PHY update procedure from a master device when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the master role maintaining a connection and initiates the PHY update procedure. IUT responds to the PHY change request and notifies the host only when a change occurs.

**Reference**

[10] 5.1.10

**Initial Condition**

Same as LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure].

**Test Procedure**

Same as LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure], except that the Lower Tester always selects a single (symmetric) PHY for both directions in the LL_PHY_UPDATE_IND PDU.

**Expected Outcome**

**Pass Verdict**

Same as LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure] except that the IUT shall only set a single (symmetric) PHY in each LL_PHY_RSP PDU.

**Inconclusive Verdict**

The IUT does not initiate at least one PHY Update Procedure during this test case because of the rules in step 4 of the test procedure in LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure].

### 4.3.4.33 LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure]

**Test Purpose**

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT’s PHY change request and the Lower Tester’s PHY change request.
The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY update procedure upon receiving the IUT’s PHY change to cause a procedure collision. The test case expects the IUT to respond to the master’s PHY update procedure after the master has rejected the IUT’s PHY update procedure.

- **Reference**
  
  [10] 5.3

- **Initial Condition**
  
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- **Test Procedure**

![Diagram of test procedure](image)

*Figure 4.196: LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure]*

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer PHY other than LE 1M, and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. Lower Tester receives an LL_PHY_REQ PDU from the IUT. Lower Tester responds with a LL_PHY_REQ with TX_PHYS and RX_PHYS set to prefer a PHY other than LE 1M and rejects the IUT’s LL_PHY_REQ PDU using a LL_REJECT_EXT_IND with ErrorCode 0x23 (LMP Error
Transaction Collision). If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.

3. The IUT may send an HCI_LE_PHY_Update_Complete event with status set to 0x23 to the Upper Tester.

4. The IUT responds to the LL_PHY_REQ from the Lower Tester with a LL_PHY_RSP with at least one bit set in each field (RX_PHYS, TX_PHYS).

5. The Lower Tester responds with an LL_PHY_UPDATE_IND PDU with M_TO_S_PHY and S_TO_M_PHY fields set to a PHY other than LE 1M and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. The Lower Tester receives a packet from the IUT acknowledging the LL_PHY_UPDATE_IND.

6. The Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.

7. At the Instant of the PHY change start maintaining the connection with the selected PHYs.

8. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements. If PHY has changed, the Lower Tester shall use the new PHYs.

9. The Upper Tester receives an HCI_LE_PHY_Update_Complete from the IUT indicating that both RX_PHY and TX_PHY match the settings used in step 2.

• Expected Outcome

Pass Verdict
The IUT transmits the LL_PHY_REQ PDU to update the selected PHYs.

If the IUT reports an HCI_LE_PHY_Update_Complete event when the Lower Tester rejects the IUT’s request, it shall have the correct error code (0x23).

The IUT responds to the LL_PHY_REQ from the Lower Tester with a LL_PHY_RSP after receiving the LL_REJECT_EXT_IND.

The test procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and adopting the new rate at the assigned event.

The IUT reports the new PHY(s) to the host with an HCI_LE_PHY_Update_Complete event.

Inconclusive Verdict
The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.34  LL/CON/SLA/BV-45-C [Protocol Timeout for PHY Update Procedure]

• Test Purpose
Test that a slave IUT terminates the Link Layer connection if the slave-initiated PHY update procedure is not completed before the procedure response timer expires.

The Lower Tester acts in the master role in the connection and ensures that the procedure initiated by the IUT is not completed.

• Reference
[10] 5.2

• Initial Condition
State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

1. Upper Tester sends an HCI_LE_Set_PHY with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M and PHY_options set to 0x0000 to the IUT and receives an HCI_Command_Status_Event.
2. Lower Tester receives an LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
3. Lower Tester acknowledges the LL_PHY_REQ PDU but does not send an LL_PHY_UPDATE_IND PDU.
4. The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero Status.
5. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

• Expected Outcome

Pass Verdict

The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in step 4 the status shall be non-zero.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).
4.3.4.35 LL/CON/SLA/BV-46-C [Handling Protocol Collision – Different Procedure – Channel Map]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT’s PHY change request and the remote device’s channel map update.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates channel map update procedure upon receiving the IUT’s PHY change to cause a procedure collision. The test case expects the IUT to respond to the master’s channel map update procedure after the master has rejected the IUT’s PHY update procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

![Test Procedure Diagram](image-url)
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. Lower Tester receives an LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.

3. Lower Tester responds with an LL_CHANNEL_MAP_IND and rejects the IUT’s LL_PHY_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.

4. The IUT sends the HCI_LE_PHY_Update_Complete event with status set to 0x2A to the Upper Tester.

5. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of channel map update.

   • Expected Outcome

     Pass Verdict
     The IUT transmits the LL_PHY_REQ PDU to update the PHY and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT’s request. The IUT properly handles the master initiated channel map request procedure.

     The test procedure is executed successfully, with the IUT acknowledging the channel map request and adopting the new channel map at the assigned event.

     The IUT maintains the connection with the Lower Tester after the channel map update completes.

     Inconclusive Verdict
     The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.36 LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

   • Test Purpose

     Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT’s PHY change request and the remote device’s connection parameters request.

     The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameters request procedure upon receiving the IUT’s PHY change to cause a procedure collision. The test case expects the IUT to respond to the master’s connection parameters request procedure after the master has rejected the IUT’s PHY update procedure.

   • Reference

     [10] 5.3

   • Initial Condition

     State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
3. Lower Tester responds with an LL_CONNECTION_PARAM_REQ and rejects the IUT’s LL_PHY_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
4. The IUT requests the Upper Tester to accept or reject the Lower Tester’s parameter update request.
5. IUT sends the HCI_LE_PHY_Update_Complete event with status set to 0x2A to the Upper Tester.
6. The Upper Tester accepts the IUT’s request to accept or reject the Lower Tester’s parameter update request.
7. The IUT responds to the LL_CONNECTION_PARAM_REQ from the Lower Tester with an LL_CONNECTION_PARAM_RSP.
8. The Lower Tester responds with an LL_CONNECTION_UPDATE_IND PDU and receives a packet from the IUT acknowledging the connection update request.

Figure 4.199: LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters]
9. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
10. At the time of the update start maintaining the connection with the new parameters.
11. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

**Pass Verdict**

The IUT transmits the LL_PHY_REQ PDU to update the selected PHYs and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT’s request.

The IUT responds positively to the master initiated connection parameters request procedure.

The test procedure is executed successfully, with the IUT acknowledging the connection update and adopting the new parameters at the assigned event.

The IUT maintains the connection with the Lower Tester after the connection update completes.

The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

**Inconclusive Verdict**

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

---

**4.3.4.37 LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update]**

- **Test Purpose**

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT’s PHY change request and the remote device’s connection update request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a connection update procedure upon receiving the IUT’s PHY change to cause a procedure collision. The test case expects the IUT to respond to the master’s connection update request procedure after the master has rejected the IUT’s PHY update procedure.

- **Reference**

[10] 5.3

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Figure 4.200: LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
3. Lower Tester responds with an LL_CONNECTION_UPDATE_IND and rejects the IUT’s LL_PHY_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
4. IUT sends the HCI_LE_PHY_Update_Complete event with status set to 0x2A to the Upper Tester.
5. Lower Tester sends empty DATA packets to the IUT, receiving acknowledgements until the event count matches the indicated time of connection update.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.

• Expected Outcome

Pass Verdict

The IUT transmits the LL_PHY_REQ PDU to update the selected PHY(s) and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT’s request.

The IUT responds positively to the master initiated connection update request procedure.

The test procedure is executed successfully, with the IUT acknowledging the connection update and adopting the new parameters at the assigned event.
The IUT maintains the connection with the Lower Tester after the connection update completes. The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

**Inconclusive Verdict**
The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

### 4.3.4.38 LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]

- **Test Purpose**
  Tests that a slave IUT follows packet time restrictions both during and after PHY change when it initiates the PHY update procedure.

  The Lower Tester, in the master role, maintains the connection.

  A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY update procedure as the Host of the IUT.

- **Reference**
  [10] 5.1.10.1

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• **Test Procedure**

![Diagram of test procedure]

---

**Figure 4.201:** LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
4. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconcusive Verdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that...
(Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.

6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and receives an HCI_Command_Complete event from the IUT in response.

7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT’s TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.

8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

9. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

10. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 1M PHY.

11. The Lower Tester receives an LL_PHY_REQ from the IUT which allows for selection of the LE 1M PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.

12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 11) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY update procedure.

13. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

14. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.

15. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

16. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

• Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in step 11) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in step 11) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

**Inconclusive Verdict**

One or more of the following:
- The IUT does not allow the Lower Tester to select the LE 2M PHY in step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in step 11).
- The IUT does not allow the Lower Tester to select the LE 1M PHY in step 11).

**4.3.4.39 LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions]**

- **Test Purpose**
  Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY update procedure from a master.

  The Lower Tester, in the master role, maintains the connection.

  A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY update procedure.

- **Reference**
  [10] 5.1.10.1

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
- **Test Procedure**

  Connection Established. IUT Slave

  ![Diagram](image)

  **Figure 4.203**: LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
4. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that
(Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.

6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and receives an HCI_Command_Complete event from the IUT in response.

7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT’s TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.

8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

9. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

10. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x01.

11. The Lower Tester receives an LL_PHY_RSP from the IUT which allows for selection of the LE 1M PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.

12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 11) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY update procedure.

13. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

14. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.

15. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

16. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

- **Expected Outcome**

  **Pass Verdict**

  All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in step 11) must have a length of 32 octets and be un-fragmented.

  All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in step 11) must have a length of less than or equal to 31 octets.

  The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

  All data packets received by the Lower Tester shall:
  - have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

**Inconclusive Verdict**

One or more of the following:
- The IUT does not allow the Lower Tester to select the LE 2M PHY in step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in step 11).
- The IUT does not allow the Lower Tester to select the LE 1M PHY in step 11).

**4.3.4.40  LL/CON/SLA/BV-51-C  [Protocol Timeout for PHY Update Procedure – No Update Request]**

- **Test Purpose**
  Test that a slave IUT terminates the Link Layer connection if master-initiated PHY update procedure is not completed before the procedure response timer expires.

  The Lower Tester acts in the master role in the connection and ensures that the procedure initiated by the Lower Tester is not completed.

- **Reference**
  
  [10] 5.2

- **Initial Condition**

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

**Figure 4.205: LL/CON/SLA/BV-51-C [Protocol Timeout for PHY Update Procedure]**

1. The Lower Tester sends an LL_PHY_REQ PDU with TX_PHYS and RX_PHYS both set to prefer a PHY other than LE 1M.
2. Lower Tester receives an LL_PHY_RSP from the IUT, and acknowledges this packet, but does not send an LL_PHY_UPDATE_IND PDU.
3. The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero status.
4. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

• Expected Outcome

**Pass Verdict**

The IUT responds to the Lower Tester’s LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHYS, RX_PHYS).

The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in step 3 the status shall be non-zero.

4.3.4.41 **LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change]**

• Test Purpose

Tests that a slave IUT follows all packet time restrictions when a PHY update procedure is initiated but no PHY change occurs.

The Lower Tester, in the master role, maintains the connection.
A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY update procedure as the Host of the IUT, but the Lower Tester does not allow it to result in a PHY change.

- Reference
  
  [10] 5.1.10.1

- Initial Condition
  
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
Test Procedure

- Connection Established. IUT Slave
  - HCI LE Set_PHY
    - ALL_PHYs = 0x03
    - Status: 0x00
  - LL_PHY_REQ
    - LL_PHY_UPDATE_IND
      - Both Directions = 1Ms/s
  - LL_PHY_REQ
    - LL_PHY_RSP
      - LL_PHY_UPDATE_IND
        - Both Directions = 2Ms/s
  - HCI LE Set Data Length
    - TxCB = 0x0148, TxCBc = 32
    - HCI Command_Complete_Event
      - Status: 0x00
  - LL_LENGTH_REQ
    - LL_LENGTH_RSP
      - RxTime = 0x0148, RxOctets = 32
  - LL_LENGTH_REQ
    - LL_LENGTH_RSP
      - RxTime = 0x0148, RxOctets = 32

IUT Initiates Data Length Procedure

- OR
  - Lower Tester initiates Data Length Procedure

UT queues data packets for remainder of the test case

- Continued in Part B...

Figure 4.206: LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
4. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an InconclusiveVerdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32.
7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT’s TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
9. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
10. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 1M PHY.
11. The Lower Tester receives an LL_PHY_REQ from the IUT in which the TX_PHYS field has the 0x01 bit set. If the TX_PHYS field does not have the 0x01 bit set, the test ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays.
sending an LL_PHY_UPDATE_IND until at least 140 octets of data are received. If the IUT does not send 140 octets of data before the procedure timeout expire, the test ends with an Inconclusive Verdict.

12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 11) must be 31 octets or less in length. If a larger packet is received during this period, the test case ends with a failed verdict.

13. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).

14. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 2M PHY.

- Expected Outcome

**Pass Verdict**

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in step 11) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in step 11) must have a length of less than or equal to 31 octets, until the Lower Tester sends the LL_PHY_UPDATE_IND in step 13).

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

**Inconclusive Verdict**

One or more of the following:

- The IUT does not allow the Lower Tester to select the LE 2M PHY in step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in step 11).
- The IUT does not set the 0x01 bit in the TX_PHYS field in step 11).
- The IUT does not send 140 octets of data before the procedure response supervision timeout expires.

4.3.4.42  **LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change]**

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY update procedure from a master but no PHY change occurs.

The Lower Tester, in the master role, maintains a connection.
A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY update procedure but does not allow it to result in a PHY change.

- Reference
  [10] 5.1.10.1

- Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Connection Established. IUT Slave

- LL_PHY_REQ
  - LL_PHY_UPDATE_IND
    - Both Directions = 1Ms/s

- LL_PHY_REQ
  - LL_PHY_RSP
    - LL_PHY_UPDATE_IND
      - Both Directions = 2Ms/s

- HCI_LE_Set_PHY
  - ALL_PHYS=0x03
  - (Status: 0x000)
  - (Optional)

- HCI_LE_Data_Length
  - TxTime = 0x0148, TxOctets = 32
  - (Status: 0x000)

- HCI_LE_Data_Length_CHANGE
  - (Optional)

- LL_LENGTH_REQ
  - LL_LENGTH_RSP
    - RxTime = 0x0148, RxOctets = 32

- Empty Data Packet
  - UT queues data packets
    - for remainder of the test case

- Data Packet
  - Length = 32 Octets, Unfragmented

- HCI_LE_Data_Packet
  - Length = 32 Octets

- Complete Procedure
  - with both directions at 2Ms/s

- IUT Initiates Data Length Procedure

- Lower Tester initiates Data Length Procedure

- UT queues data packets
  - for remainder of the test case

- Continued in Part B...

Figure 4.208: LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
4. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and receives an HCI_Command_Complete event from the IUT in response.
7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT’s TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
9. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
10. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x03.

11. The Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 1M PHY bit must be set in the TX_PHYS field by the IUT or the test case ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 140 octets are received.

If the IUT does not send 140 octets of data before the procedure timeout expire, the test ends with an Inconclusive Verdict.

12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 11 must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict.

13. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).

• Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in step 11 must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in step 11 must have a length of less than or equal to 31 octets until the Lower Tester sends the LL_PHY_UPDATE_IND in step 13.

All data packets received by the Lower Tester shall:
- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:
- The IUT does not allow the Lower Tester to select the LE 2M PHY in step 4.
- The IUT specifies values of TtTime < 0x0148 or TxOctets < 32 in step 7.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in step 11.
- The IUT does not set the LE 1M PHY bit in the TX_PHY field in step 11.
- The IUT does not send 140 octets of data before the procedure response supervision timeout expires.

4.3.4.43 LL/CON/SLA/BV-54-C [Slave Receiving Data, LE Coded, CI Change]

• Test Purpose

Test that a slave IUT is able to receive data from a master device when the master is transitioning between 125kbit and 500kbit coded rates. Confirm that IUT responds within the allowed T_IFS times
for each packet at either coded rate. Test is performed with the IUT’s minimum and maximum supported packet lengths. A Data Length Update Procedure is performed if required.

The Lower Tester acts in the master role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- **Reference**
  
  [10] 4.5

- **Initial Condition**
  
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, maximum supported connection interval, common slave latency, common timeout, common channel map, any SCA value).

- **Test Procedure**
  
  Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE Coded PHY.

4. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE Coded PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select LE Coded PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE Coded PHY.

If (TSPX_TxOctets_Max > 27) OR (TSPX_TxTime_Max > 328) then the Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to TSPX_TxTime_Max and TxOctets set to TSPX_TxOctets_Max and receives an HCI_Command_Complete event from the IUT in response, otherwise go to step 8.

If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max. If the IUT does not initiate a Data Length Update Procedure TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max and receives a response from the IUT.

Configure Lower Tester to send 288 data packets of length connEffectiveMaxTxOctets with all payload octets set to continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.). For each packet sent the CI field will alternate in the pattern below. Each number in the pattern represents the CI field for that particular packet to be sent.

```
2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 8 2 8 8 8 8 8 8 8
2 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 8 2 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
8 8 8 2 2 8 8 8 8 2 8 8 8 8 8 8 8
8 8 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8
2 8 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8
2 8 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8
8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8 8
2 8 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8
2 8 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8
8 8 8 2 2 8 8 8 8 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
```

Lower Tester sends one or more DATA packets to the IUT, using the acknowledgement scheme and the data channel selection parameters. Lower Tester receives an empty DATA packet in response from the IUT for each DATA packet sent by the Lower Tester, received with acceptable T_IFS timing (150 µs +/- 2 µs).
10. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing one or more data elements sent in step 8. The received data should be in the form of continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.).

11. Repeat steps 8–10 until all data to be sent has been reported to the Upper Tester.

12. Repeat the entire data exchange procedure again from steps 8–11, but utilizing the minimum data payload size (27 bytes) instead of connEffectiveMaxTxOctets for each data packet.

• Expected Outcome

Pass Verdict
The test procedure completes with the IUT acknowledging all the data sent.

The IUT reports all data correctly with HCI_LE_Data_Packet events containing continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.).

All responses from the IUT are received with acceptable T_IFS timing (150 µs +/- 2 µs).

4.3.4.44 LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]

• Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it initiates the PHY Update Procedure. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester, in the master role, maintains a connection.

A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY Update Procedure as the Host of the IUT.

• Reference

[10] 5.1.10.1

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Connection Established. IUT Slave

IUT Initiates Data Length Procedure

Lower Tester Initiates Data Length Procedure

UT queues data packets for remainder of the test case

Figure 4.211: LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.

4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure,
the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.

5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHYS and TX_PHYS both set to prefer LE Coded PHY.

8. The Lower Tester receives an LL_PHY_REQ from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions, the test case ends with an Inconclusive Verdict.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 8 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY Update Procedure.

10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.

12. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.

13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

• Expected Outcome

   **Pass Verdict**

   All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in step 8 must have a length of 141 octets and be un-fragmented.

   During step 10 and 11, the IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

   All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

   For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.
Inconclusive Verdict

One of more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in step 8.
- The IUT does not allow the Lower Tester to select the LE Coded PHY in step 8.
- The IUT sends autonomously an LL_LENGTH_REQ after the instant specified in the LL_PHY_UPDATE_IND in step 8.

4.3.4.45   LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded]

• Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY Update Procedure from a master. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester, in the master role, maintains a connection.

A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure.

• Reference

[10] 5.1.10.1

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Connection Established. IUT Slave

ICT Initiates Data Length Procedure

Empty Data Packet

Data Packet
Length = 141 Octets, Unfragmented

Continued in Part B...

Figure 4.213: LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.
4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure,
the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.

5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with a fail verdict.

7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.

8. The Lower Tester receives an LL_PHY_RSP from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 8 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY Update Procedure.

10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.

12. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.

13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

- Expected Outcome

  **Pass Verdict**

  All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in step 8 must have a length of 141 octets and be un-fragmented.

  All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in step 8 must have a length of less than or equal to 140 octets.

  The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

  All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

  For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.
Inconclusive Verdict

One or more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in step 8.
- The IUT does not allow the Lower Tester to select the LE Coded PHY in step 8.
- The IUT sends autonomously an LL_LENGTH_REQ after the instant specified in the LL_PHY_UPDATE_IND in step 8.

4.3.4.46 LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded]

- Test Purpose
  Tests that a slave IUT still transmits data even when the TxTime and/or RxTime values for LE Coded PHY suggest a smaller possible data length than the minimum length data PDU (27 octets).
  The Lower Tester, in the master role, maintains a connection.
  If a Data Length Update Procedure is performed, minimum settings are used. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure.

- Reference
  [10] 5.1.9

- Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Connection Established. IUT Slave

- HCI_LE_Set_PHY
  - ALL_PHYS=0x00
  - (Status: 0x00)
  - HCI_LE_PHY_Update_Complete_Event

- LL_LENGTH_REQ
- LH_Length_RSP
  - RxTime = 0x0148, RxOctets = 27
  - HCI_LE_Data_Length_Change_Event
  - (Optional)

- IUT Initiates Data Length Procedure

- Empty Data Packet
- Data Packet(s)
  - Length <= 27 Octets

- UT queues data packets for remainder of the test case

- Continued in Part B...

Figure 4.215: LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Optionally, if the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 328 and RxOctets set to 27.
4. The Upper Tester begins to queue data packets to the IUT with a length of 27 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
5. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.
6. The Lower Tester receives an LL_PHY_RSP from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.

7. The Lower Tester expects that the data packets received are less than or equal to 27 octets in length.

8. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 27 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

9. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.

10. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY and, if LE Data Packet Length Extension feature is supported by the IUT, an HCI_LE_Data_Length_Change event with MaxTxOctets and MaxRxOctets set to 27 and MaxTxTime and MaxRxTime set to 2704. These events may be in either order.

11. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are less than or equal to 27 octets in length. This data exchange continues for at least 10 connection events after the Instant.

12. Repeat steps 5–11, except that in step 5, change 0x04 to 0x01; in steps 6, 9, and 10, change the LE Coded PHY to the LE 1M PHY; and in step 10, change 2704 to 328.

• Expected Outcome

Pass Verdict

Data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP (in steps 6 and 11) must have a length of 27 octets or less.

The IUT must transition to the LE Coded PHY and LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

If LE Data Packet Length Extension feature is supported by the IUT, the Upper Tester receives HCI_LE_Data_Length_Change events in step 10 with the correct values.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

- The IUT does not allow the Lower Tester to select the required PHY in step 6.
- The IUT sends autonomously an LL_LENGTH_REQ after the instant specified in the LL_PHY_UPDATE_IND in step 6.
4.3.4.47  LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]

- **Test Purpose**
  Tests that a slave IUT follows all packet time restrictions when a PHY Update Procedure is initiated but no PHY change occurs. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant, even if no change occurs.

  The Lower Tester acts in the master role maintaining a connection.

  A Data Length Update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY Update Procedure as the Host of the IUT.

- **Reference**
  [10] 5.1.10.1

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

Figure 4.217: LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.
4. If the IUT initiates a Data Length Update procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update procedure, the Lower Tester shall initiate the Data Length Update procedure with RxTime set to 0x04D8 and RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHY and TX_PHY both set to prefer the LE Coded PHY.
8. The Lower Tester receives an LL_PHY_REQ from the IUT in which the TX_PHY field has the 0x04 bit set. If the TX_PHY field does not have the 0x04 bit set, the test ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 423 octets of data are received.
9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 8 must be 140 octets or less in length. If a larger packet is received during this period, the test case ends with a failed verdict.
10. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).
11. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

- **Expected Outcome**

  **Pass Verdict**

  All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in step 8) must have a length of 141 octets and be un-fragmented.

  All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in step 8) must have a length of less than or equal to 140 octets, until the Lower Tester sends the LL_PHY_UPDATE_IND in step 10.

  All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

  For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

- **Inconclusive Verdict**

  One or more of the following:

  - The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
  - The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in step 8.
  - The IUT does not allow the Lower Tester to select the LE Coded PHY in step 8.

**4.3.4.48 LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]**

- **Test Purpose**

  Tests that a slave IUT follows all packet time restrictions both during and after PHY change when it responds to a PHY Update Procedure from a master device but no PHY change occurs. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant, even when no change occurs.

  The Lower Tester, in the master role, maintains a connection. A Data Length Update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure but does not allow it to result in a PHY change.

- **Reference**

  [10] 5.1.10.1

- **Initial Condition**

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave ConnSlaveLatency_MAX, selected timeout, any SCA value).
Test Procedure

- Connection Established. IUT Slave

IUT Initiates Data Length Procedure

LL_PHY_REQ
- LL_PHY_UPDATE_IND
  - Both Directions = 1Ms/s

Optional

HCI_LE_Set_PHY
- ALL_PHYS=0x03
  - (Status: 0x00)

Both Directions = 1Ms/s

IUT Initiates Data Length Procedure

LL_LENGTH_REQ
- LL_LENGTH_RSP
  - RxTime = 0x04D8, RxOctets = 141

 HCI_LE_Data_Length_Change_Event
  - (Optional)

Lower Tester initiates Data Length Procedure

LL_LENGTH_REQ
- LL_LENGTH_RSP
  - RxTime = 0x04D8, RxOctets = 141

 HCI_LE_Data_Length_Change_Event
  - (Optional)

UT queues data packets
for remainder of the test case

Empty Data Packet
- Data Packet
  - Length = 141 Octets, Unfragmented

Continued in Part B...

HCI_LE_Data_Packet
- Length = 141 Octets

HCI_Command_Complete_Event
- (Status: 0x00)

HCI_Command_Status_Event
- (Status: 0x00)

Figure 4.219: LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.

4. If the IUT initiates a Data Length Update procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update procedure, the Lower Tester shall initiate the Data Length Update procedure with RxTime set to 0x04D8 and RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.

5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x05.

8. The Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE Coded PHY bit must be set in the TX_PHYS field by the IUT or the test case ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 423 octets are received.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 8 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict.
10. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).

- **Expected Outcome**

  **Pass Verdict**

  All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in step 8 must have a length of 141 octets and be un-fragmented.

  All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in step 8 must have a length of less than or equal to 140 octets until the Lower Tester sends the LL_PHY_UPDATE_IND in step 10.

  All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

  For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

  **Inconclusive Verdict**

  One or more of the following:

  - The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
  - The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in step 8.
  - The IUT does not set the LE Coded PHY bit in the TX_PHY field in step 8.

4.3.4.49 **LL/CON/SLA/BV-77-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]**

This test is identical to **LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]** except that the IUT is in the Slave role.

4.3.4.50 **LL/CON/SLA/BV-78-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]**

This test is identical to **LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]** except that the IUT is in the Slave role.

4.3.4.51 **LL/CON/SLA/BV-79-C [Slave Data Length Update – Master does not support; LE 1M PHY]**

This test is identical to **LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY]** except that the IUT is in the Slave role.

4.3.4.52 **LL/CON/SLA/BV-80-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]**

This test is identical to **LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]** except that the IUT is in the Slave role.
4.3.4.53 LL/CON/SLA/BV-81-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]

This test is identical to LL/CON/MAS/BV-77-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY] except that the IUT is in the Slave role.

4.3.4.54 LL/CON/SLA/BV-82-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY]

This test is identical to LL/CON/MAS/BV-78-C [Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY] except that the IUT is in the Slave role.

4.3.4.55 LL/CON/SLA/BV-83-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY]

This test is identical to LL/CON/MAS/BV-79-C [Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY] except that the IUT is in the Slave role.

4.3.4.56 LL/CON/SLA/BV-84-C [Slave Data Length Update – Master does not support; LE Coded PHY]

This test is identical to LL/CON/MAS/BV-80-C [Master Data Length Update – Slave does not support; LE Coded PHY] except that the IUT is in the Slave role.

4.3.4.57 LL/CON/SLA/BV-85-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]

• Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when a feature exchange has not been performed and the remote device does not support the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and sends an LL_UNKNOWN_RSP PDU in response to the IUT’s request.

• Reference

[3] 5.1.7

• Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
• Test Procedure

1. If the IUT autonomously initiates a feature exchange before step 3, the test ends with an Inconclusive Verdict.

2. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.

3. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester) and the Lower Tester responds with an LL_UNKNOWN_RSP PDU with the Opcode field set to LL_CONNECTION_PARAM_REQ (0x0F).

4. Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the error code Unsupported Remote Feature (0x1A).

5. Upper Tester resends the same HCI_LE_Connection_Update command to the IUT and expects the IUT to respond with either:
   a. an HCI_Command_Status event containing the error code Unsupported Remote Feature (0x1A), or
   b. an HCI_Command_Status event indicating success and an HCI_LE_Connection_Update_Complete event containing the error code Unsupported Remote Feature (0x1A).

If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.
• Expected Outcome

Pass Verdict
The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update the connection parameters.
The IUT reports an HCI_LE_Connection_Update_Complete event containing the error code 
Unsupported Remote Feature (0x1A).
The IUT rejects the second command with the Unsupported Remote Feature (0x1A) error code.

4.3.4.58 LL/CON/SLA/BV-86-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

• Test Purpose
Test that a slave IUT is able to reject the connection parameter request procedure after the feature exchange reveals that the remote device does not support the request.
The Lower Tester acts in the master role maintaining a connection and initiates feature exchange, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT.

• Reference
[3] 5.1.7

• Initial Condition
Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.
State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).
1. Lower Tester initiates a feature exchange, unless the IUT has already done so. Lower Tester indicates that it does not support the Connection Parameter Request Procedure.

2. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects the IUT to respond with either:
   a. an HCI_Command_Status event containing the error code Unsupported Remote Feature (0x1A), or
   b. an HCI Command Status event indicating success and an HCI_LE_Connection_Update_Complete event containing the error code Unsupported Remote Feature (0x1A).

   If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.

• Expected Outcome
  
  Pass Verdict
  
  The IUT rejects the command with the Unsupported Remote Feature (0x1A) error code.

4.3.4.59 LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]

• Test Purpose
  
  Test that a slave IUT accepts the master transmission at the beginning of an event as the anchor point, irrespective of the checksum result.

  The Lower Tester acts in the master role, starting all events after connection setup with packets with invalid checksums and observes the responses and packet contents from the IUT.

• Reference
  
  [3] 4.5.5
• **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, Lower Tester address, common connection interval, common slave latency, common channel map, any SCA value).

• **Test Procedure**

Execute the test procedure using a connection timeout value of 3000ms. The test uses the common current and next variables for SN and NESN.

![Diagram of test procedure](image)

*Figure 4.223: LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]*

1. Lower Tester sends a DATA packet with an invalid checksum once a connection interval to the IUT, using the data channel selection parameters.
2. Lower Tester receives a DATA packet after T_IFS in events where the IUT is required to listen. Allow mismatches for packets not received correctly.
3. Repeat steps 1–2 for a period longer than the connection supervision timeout value:
4. Upper Tester receives an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating connection timeout (0x08).

• **Test Condition**

The parameters in this test are calculated for a BER of 0.1 percent or better.

• **Expected Outcome**

Pass Verdict

The test procedure executes with the parameters selected,

The IUT responds in at least 65 of the connection events measured (300).

• **Notes**

The error rate for the preamble and access address for the BER is the same as for test LL/CON/SLA/BV-10-C [Accepting Parameter Update], but the response rate required is divided by slave latency plus 1.
4.3.4.60  LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]

- **Test Purpose**
  Test the correct behavior of a slave IUT when TERMINATE_IND packets are not acknowledged. The Lower Tester acts in the master role, receiving TERMINATE_IND packets from the IUT and not acknowledging them.

- **Reference**
  [3] 5.1.6

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value)

- **Test Procedure**
  Execute the test procedure using the common connection parameters.

  ![Diagram](image)

  **Figure 4.224: LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]**

  1. **Upper Tester** sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps' execution and receives an HCI_Command_Status in response.
  2. **Lower Tester** expects the IUT to respond to a master transmission with a TERMINATE_IND packet. Do not acknowledge the termination or any following packet but continue master transmissions.
  3. **Lower Tester** sends an empty DATA packet to the IUT not acknowledging any responses. Repeat until T_Terminate timer expires.
  4. **Upper Tester** receives an HCI_Disconnection_Complete event including status of 0x00 (success), and a reason code of 0x16 ("Connection Terminated by Local Host") or 0x22 ("LL Response Timeout") from the IUT indicating loss of the link.
Expected Outcome

Pass Verdict

The IUT reports the connection termination with an HCI event.

The IUT keeps sending TERMINATE_IND packets until T_Terminate timer expires.

4.3.4.61 LL/CON/SLA/BI-04-C [Rejecting Connection Change]

Test Purpose

Test that a slave IUT either terminates the connection or successfully maintains the connection upon receiving a control packet indicating a past event for the change.

The Lower Tester acts in the master role in the connection, transmitting control packets with invalid event counter values and observes the IUT notifying that the connection has been terminated.

Reference

[3] 5.1.1, 5.1.2

Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value).

IUT is Slave and Lower Tester is Master.

Test Procedure

Execute the test procedure using the common connection parameters.
1. Lower Tester sends an LL_CONNECTION_UPDATE_IND packet setting the connection parameters to the intermediate values, but the event counter equal to or less than the present event. The IUT may send a packet acknowledging the connection update request.

2. Upper Tester receives an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating instant passed (0x28).

3. A connection is established.

4. Lower Tester sends an LL_CHANNEL_MAP_IND packet with the event counter equal to or less than the present event to the IUT using the first selected channel map.
   
   **Alternative 1:**
   - The IUT may send an acknowledgement.
   - The Upper Tester receives an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating instant passed (0x28).

   **Alternative 2:**
   - The IUT sends an acknowledgement.
   - The IUT does not send an HCI_Disconnection_Complete event to the Upper Tester and maintains the connection with the Lower Tester.

---

**Figure 4.225: LL/CON/SLA/BI-04-C [Rejecting Connection Change]**
• Expected Outcome
  
  **Pass Verdict**
  
  If alternative 1 is followed in step 4:
  
  - The IUT stops maintaining the connection, at any time before a period equal to the connection supervision timeout value has passed from the invalid request.
  
  - The IUT reports the connection failure with an HCI event.
  
  If alternative 2 is followed in step 4, the IUT maintains the connection with the Lower Tester and does not report a connection failure.

4.3.4.62 LL/CON/SLA/BI-05-C [Slave Connection Control Timer]

• Test Purpose
  
  Test that a slave device is able to recover from a control procedure failure.
  
  The Lower Tester acts in the master role, starting and interrupting a control procedure in order for the slave connection control timer to expire.
  
  The host shall be informed about the failure of a control procedure if the link is not disconnected before the completion of the control procedure.
  
  The host should not be informed about the failure of a control procedure if the link is disconnected before the completion of the control procedure.

• Reference
  
  [3] 5.2

• Initial Condition
  
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value)

• Test Procedure
  
  Execute the test procedure using the common connection parameters.
1. Once connection is established, if the IUT has sent LL_VERSION_IND packet by itself, the Lower Tester acknowledges the version packet. The Lower Tester continues the master transmissions but it never sends the LL_VERSION_IND packet. Then, continue on step 6.

2. If the IUT does not send LL_VERSION_IND packet by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT containing the connection handle from the preamble steps’ execution and receives an HCI_Command_Status_Event in response.

3. The Lower Tester expects the IUT to respond to a master transmission with a VERSION_IND packet. The Lower Tester acknowledges the version packet.

4. The Lower Tester sends empty DATA packets to the IUT until the connection control timeout value, or until step 5 executes.

5. The IUT sends to the Upper Tester the event
   HCI_Read_Remote_Version_Information_Complete_Event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from LL response timeout.

6. The Upper Tester receives an HCI_Disconnection_Complete_Event from the IUT indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from connection control transaction timeout.

- Expected Outcome

Pass Verdict
The test procedure executes successfully, with the IUT stopping to respond to the master transmissions.

The IUT reports the connection termination with an HCI event.
4.3.4.63  LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]

• Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device does not respond to the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and does not respond to the IUT’s request.

• Reference

[3] 5.1.7

• Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

• Test Procedure

![Diagram of test procedure]

Figure 4.227: LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). The Lower Tester acknowledges the
LL_CONNECTION_PARAM_REQ PDU but does not respond to the LL_CONNECTION_PARAM_REQ.

3. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

- **Expected Outcome**

  **Pass Verdict**

  The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

4.3.4.64  **LL/CON/SLA/BI-08-C [Accepting Connection Parameter Request – Illegal Parameters]**

- **Test Purpose**

  Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master contains illegal parameters.

- **Reference**

  [3] 5.1.7

- **Initial Condition**

  Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- **Test Procedure**

  1. **Lower Tester** sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection interval min and max to outside the valid range i.e., 4 (6ms).

  2. **Lower Tester** receives an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1E.
Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester’s request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode.

4.3.4.65 LL/CON/SLA/BI-09-C [Responding to PHY Update Procedure – Instant In Past]

Test Purpose

Tests a slave IUT’s ability to cope when a master-initiated PHY update procedure specifies an instant that is in the past.

Reference

[10] 5.1.10

Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

Test Procedure
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the ALL_PHYS fields set to a value of 0x03. Upper Tester receives an HCI_Command_Status event AND AN HCI_PHY_UPDATE_COMPLETE EVENT indicating success in response.

2. The Lower Tester sends an LL_PHY_REQ PDU with TX_PHYS and RX_PHYS both set to prefer a PHY other than LE 1M.

3. Lower Tester receives an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). A PHY other than LE 1M must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.

4. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select a PHY other than LE 1M and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be greater than or equal to 32767 and completes the procedure.

   Alternative 1:
   - The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero status.
   - The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x28) to the Upper Tester and the IUT stops maintaining the connection.

   Alternative 2:
   - The IUT sends an HCI_LE_PHY_Update_Complete with TX_PHYS and RX_PHYS both set to the PHYs specified in the LL_PHY_UPDATE_IND PDU.
   - The IUT maintains the connection with the Lower Tester on those PHYs.

• Expected Outcome

  Pass Verdict
  The IUT responds to the Lower Tester’s LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHYS, RX_PHYS).

  If alternative 1 is followed in step 4:
  - If the LL_PHY_UPDATE_IND M_TO_S_PHY or S_TO_M_PHY field is non-zero, the IUT sends an HCI_Disconnection_Complete_Event (Reason 0x28) and the IUT stops maintaining the connection.
  - If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in step 5 the status shall be non-zero.

  If alternative 2 is followed in step 4, the IUT sends an HCI_LE_PHY_Update_Complete with PHYS other than LE 1M, maintains the connection with the Lower Tester and does not report a connection failure.

  Inconclusive Verdict
  The IUT does not select a PHY other than LE 1M in both fields in step 3).
4.3.5 MAS

Tests that the IUT behaves according to the connection procedures in the master role.

4.3.5.1 LL/CON/MAS/BV-02-C [Events with Slave Latency]

- Test Purpose
  Test that a master IUT is able to maintain a connection when the slave using the slave latency mechanism.
  The Lower Tester acts as a slave and uses latencies up to the maximum supported by the IUT.

- Reference
  [3] 4.5

- Initial Condition
  Parameters: LL_master_connSlaveLatency_MAX
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, up to LL_master_connSlaveLatency_MAX, timeout, common channel map, not encrypted)

- Test Procedure
  Execute the test procedure using a connection interval of 32 ms, a latency of 5 events and a connection timeout parameter of 32 s.
1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0xFFFFFFFF, for a data total length of 4, with Packet_Boundary_Flag flag set.

2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.

3. Lower Tester sends a DATA packet with in events required by the slave latency parameter only, T_IFS after the packet from the IUT. Use only the initial values for SN and NESN.

4. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 3 and with the Packet_Boundary_Flag flag set.

5. Expect the IUT to continue to repeat the acknowledgement or to indicate that no response was received, with the SN matching the current NESN and the NESN matching the current SN until the Lower Tester sends a DATA packet after slave latency parameter. In case of flow control allow the IUT to updated SN without updating NESN in sent packet.

6. Repeat steps 1–5 600 times.
7. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

• Expected Outcome

  Pass Verdict
  The IUT maintains the connection with slave latency in use,
  The IUT observes the acknowledgement scheme taking into account possible flow control in the procedure steps,
  The IUT clock drift between the events where the slave is active is acceptable.

• Notes
  The connection parameters are selected such that the probability that a connection will be terminated by the connection supervision timer expiring when the slave is using the maximum latency is close to zero. Only the shorter connection interval is tested with slave latency below the maximum possible. When the connection supervision timer is restricted only to \((1 + \text{connSlaveLatency}) \times \text{connInterval}\) as in the connection setup parameter constraints, there is still about 0.5 percent probability that a connection terminates, which is expected to occur in a geometric distribution about once every 170 tries which is not acceptable for a repeatable test. The probabilities are calculated for empty data packets.

4.3.5.2 LL/CON/MAS/BV-03-C [Master Sending Data]

• Test Purpose
  Test that a master IUT is able to send data to a slave device.
  The Lower Tester acts in the slave role in the connection, submits data for the master to transmit and observes the data in the packets from the master.

• Reference
  [3] 4.5

• Initial Condition
  Parameters: \(\text{LL_master_payload_length_MIN}, \text{LL_master_payload_length_MAX}\)
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• Test Procedure
  Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.
1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.

2. Upper Tester receives the appropriate HCI_Number_Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.

3. Lower Tester receives a DATA packet from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1. Lower Tester sends an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel.

4. Repeat steps 2–3 until all elements submitted in step 1 have been received.

5. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x01 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) is successfully submitted.

6. Upper Tester receives the appropriate HCI_Number_Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.

7. Lower Tester receives a DATA packet from the IUT, with the LLID field set to 0x01 and containing a data element submitted in step 5. Send an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel.
8. Repeat steps 6–7 until all elements submitted in step 5 have been received.

9. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, with Packet_Boundary_Flag flag equal to 0x00 in the first packet and equal to 0x01 in the following until the selected number of octets (1000) are successfully submitted.

10.Upper Tester receives the appropriate HCI_Number_Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.

11. Lower Tester receives a DATA packet from the IUT, containing a data element submitted in step 9: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following.

12. Lower Tester responds with a DATA packet using the acknowledgement scheme to the IUT on the same data channel.

13. Repeat steps 10–12 until all data sent in step 9 have been reported.

Optional, for devices with a data_packet_length (as defined in Figure 4.4: Buffer Size Read Preamble Steps) over 27 octets.

14. Optional. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 and data elements with the value 0x00, using a data total length from 28 to the data_packet_length (as defined in Figure 4.4: Buffer Size Read Preamble Steps), until the selected number of octets (1000) are successfully submitted. It receives the appropriate HCI_Number_Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.

15. Optional. Lower Tester receives a DATA packet from the IUT, containing a data element submitted in step 10: with the LLID field set to 0x00. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.

16. Optional. Repeat step 15 until all elements submitted in step 14 have been received.

17. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble steps).

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending all of the data and maintaining the correct sequence of the fragmentation flags, as specified in Section 4.1.6.

• Notes

The number of octets should correspond to a typical data length transmitted by a Host, here set to 1000.

4.3.5.3 LL/CON/MAS/BV-04-C [Master Receiving Data]

• Test Purpose

Test that a master IUT is able to receive data from a slave device.

The Lower Tester acts in the slave role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

• Reference

[3] 4.5
• **Initial Condition**

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

• **Test Procedure**

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.

![Diagram of test procedure](image-url)

**Figure 4.232: LL/CON/MAS/BV-04-C [Master Receiving Data]**

1. Configure Lower Tester to send 1000 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
2. Lower Tester receives a DATA packet from the IUT and sends a DATA packet in response using the acknowledgement scheme to the IUT on the same data channel. Repeat until the selected number of octets (1000) is acknowledged by the IUT.
3. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 2 and with the Packet_Boundary_Flag flag set to 0x02. Repeat until the selected number of octets (1000) is received in HCI_LE_Data_Packet event.
4. Configure Lower Tester to send 1000 data packets with the LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.
5. Repeat step 2.
6. Upper Tester receives HCI_LE_Data_Packets from the IUT containing the data sent in step 5 and with the Packet_Boundary_Flag flag set to 0x01. Repeat until the selected number of octets (1000) is received in HCI_LE_Data_Packet event.

7. Configure Lower Tester to send one data packet with LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00. After, send 990 data packets with LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.

8. Repeat step 2.

9. Upper Tester receives HCI_LE_Data_Packets from the IUT containing the data sent in step 8. Repeat until the selected number of octets (1000) is received in HCI_LE_Data_Packet event. First HCI_LE_Data_Packet has the Packet_Boundary_Flag flag set to 0x02 and rest of HCI_LE_Data_Packet has the Packet_Boundary_Flag flag set to 0x01.

10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble steps).

• Expected Outcome

  Pass Verdict

  The test procedure completes with the IUT acknowledging all the data sent,

  The IUT reports the data correctly using HCI_Data_Packet events and the HCI fragmentation flags, as specified in Section 4.1.6.

4.3.5.4 LL/CON/MAS/BV-05-C [Master Sending and Receiving Data]

• Test Purpose

  Test that a master IUT is able to send and receive data to/from a slave device.

  The Lower Tester acts in the slave role in the connection, both submits data from the host of the IUT to transmit and sends data to the IUT according to the acknowledgement scheme and observes the data received and reported to the host of the IUT.

• Reference

  [3] 4.5

• Initial Condition

  Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• Test Procedure

  Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.
1. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 in the first packet and to 0x01 in the following and using a payload length of 10 with the payload octets set to 0xFF.

2. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, with Packet_Boundary_Flag flag set to 0x00 in the first packet and set 0x01 in the following until the selected number of octets (1000) are successfully submitted.

3. Expect an HCI_Number_Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.

4. Expect a DATA packet from the IUT, containing a data element submitted in 1: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following.

5. Respond with a DATA packet using the acknowledgement scheme to the IUT on the same data channel.

6. Repeat steps 4–5 until all data sent in step 2 have been reported.

7. Upper Tester receives HCI_LE_Data_Packets from the IUT containing the data sent in step 5 with the Packet_Boundary_Flag flag set to 0x02 in the first packet and set to 0x01 in the following. Payload octets shall be 0xFF.

- Expected Outcome

  **Pass Verdict**

  The test procedure completes with the IUT acknowledging all the data sent and reporting all data received, as specified in Section 4.1.6.

4.3.5.5 **LL/CON/MAS/BV-07-C [Requesting Parameter Update]**

- Test Purpose

  Test that a master IUT is able to perform the connection parameter update procedure.

  The Lower Tester acts in the slave role maintaining a connection and submits HCI commands to start connection parameter update as the host of the IUT, and then observes the procedure carried out by the IUT.

- Reference

  [3] 5.1.1
• **Initial Condition**

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• **Test Procedure**

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT, including parameters to use maximum connection interval and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives a CONNECTION_UPDATE_IND control packet from the IUT, with the parameters submitted in 1. Send an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel. Repeat until the IUT has sent the packet following the connection update request, or up to a number of events (15):
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet:
4. Once the event count matches the time, the new parameters will be in use and the time between the data packet and the next empty packet sent by the IUT is one new connection interval (considering jitter).
5. Upper Tester receives an HCI_LE_Connection_Update_Complete event after the time of the update from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

The IUT transmits the request to update connection parameters.
The IUT maintains the connection with the new parameters in use.
The IUT reports the new parameters with an HCI event.

4.3.5.6 LL/CON/MAS/BV-08-C [Master Sending Termination]

• Test Purpose

Test that a master IUT is able to terminate a connection by sending the termination packet.
The Lower Tester acts in the slave role in the connection, on receiving the termination packet from the IUT, acknowledges it and observes the master stopping transmissions.

• Reference

[3] 5.1.6

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• Test Procedure

![Diagram of test procedure]

1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps’ execution and receives an HCI_Command_Status event in response.
2. Lower Tester expects the IUT to transmit a TERMINATE_IND packet. Lower Tester acknowledges the termination packet with an empty DATA response packet in the same event.
3. Lower Tester receives no master transmissions from the IUT up to a time equal to the connection supervision timeout:
4. Interleave with step 3: Upper Tester receives an HCI_Disconnection_Complete event from the IUT indicating that the connection termination procedure requested in step 1 was successful.

Figure 4.235: LL/CON/MAS/BV-08-C [Master Sending Termination]
• Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT sending termination packets until acknowledgement from Lower Tester.

The IUT stops the master transmissions after the termination packets have been acknowledged.

The IUT reports the connection termination with an HCI event.

4.3.5.7 LL/CON/MAS/BV-09-C [Master Accepting Termination]

• Test Purpose

Test that a master IUT accepts connection termination by a slave sending the termination packet.

The Lower Tester acts in the slave role in the connection, sends the termination packet to the IUT and observes the master acknowledgement and stopping transmissions.

• Reference

[3] 5.1.6

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• Test Procedure

![Diagram of test procedure](image)

**Figure 4.236: LL/CON/MAS/BV-09-C [Master Accepting Termination]**

1. Lower Tester receives an empty DATA packet and transmits a TERMINATE_IND packet to the IUT in response. Expect an acknowledgement from the IUT in the following event.
2. Lower Tester receives no master transmissions from the IUT up to a time equal to the connection supervision timeout:
3. Interleave with step 2: Upper Tester receives an HCI_Disconnection_Complete_event from the IUT indicating termination requested by the peer device and containing the connection handle from the preamble.
• Expected Outcome

Pass Verdict
The test procedure executes successfully, with the IUT acknowledging the termination packet before $T_{\text{Terminate}}$ timer expires.
The IUT stops the master transmissions after it has acknowledged the TERMINATE_IND packet.
The IUT reports the connection termination with an HCI event.

4.3.5.8 LL/CON/MAS/BV-10-C [Master Supervision Timer]

• Test Purpose
Test that a master IUT terminates a connection by the supervision timer.
The Lower Tester acts in the slave role in the connection, stops responding in events, and then observes the IUT stopping transmissions and reporting the connection termination.

• Reference
[3] 4.5.2

• Initial Condition
Parameters: LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, 0 slave latency, selected timeout, common channel map, not encrypted)

• Test Procedure

Figure 4.237: LL/CON/MAS/BV-10-C [Master Supervision Timer]
1. The connection has been established with the maximum connection supervision timeout and connection interval values supported and use zero slave latency.
2. Lower Tester stops responding to the master transmissions after a specific event and until connection timeout expires.
3. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, containing the connection handle and termination by timeout as the reason after a time no shorter than the connection supervision timeout value from the last event responded in.
4. Establish the connection with the minimum connection supervision timeout and connection interval values supported and use zero slave latency.
5. Repeat steps 2–3.

• Expected Outcome

   Pass Verdict

   The test procedure executes successfully, with the IUT reporting the connection termination event after the selected connSupervisionTimeout value.

4.3.5.9 LL/CON/MAS/BV-13-C [Feature Setup Request]

• Test Purpose

   Test that a connected master IUT requests and performs the feature setup procedure, activating the correct features.

   The Lower Tester acts in the slave role in a maintained connection and responds to the request from the IUT to perform feature setup.

• Reference

   [3] 4.6, 5.1.4

• Initial Condition

   Parameters: Parameters: LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

   State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, selected slave latency, selected timeout, common channel map, not encrypted)
• Test Procedure

1. Upper Tester sends an HCI_LE_Read_Local_Supported_Features command and receives an HCI_Command_Complete_Event with the correct feature set value.
2. Upper Tester sends an HCI_LE_Read_Remote_Features command.
3. The IUT sends an LL_FEATURE_REQ PDU including the configured feature set or may have autonomously sent an LL_FEATURE_REQ prior to receiving the HCI command from the Upper Tester. In either case the IUT waits for an LL_FEATURE_RSP PDU from the Lower Tester.
4. The IUT sends the HCI_LE_Read_Remote_Features_Complete event to the Upper Tester.

• Expected Outcome

Pass Verdict
All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT requesting the feature information and acknowledging the reply.

The IUT reports the feature setup procedure completed with an HCI event containing the correct used features for the connection.

• Notes

The command called LE Read Remote Features and the event called LE Read Remote Features Complete in this test and the 5.0 Core Specification were called LE Read Remote Used Features and LE Read Remote Used Features Complete in the 4.0, 4.1, and 4.2 Core Specifications.

4.3.5.10 LL/CON/MAS/BV-14-C [Master Retransmission Request]

• Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme.

The Lower Tester acts in the slave role, using invalid checksums to prompt a repeated retransmission request from the IUT.
• Reference
  [3] 4.5.9

• Initial Condition
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

• Test Procedure
  Execute the test procedure using the common connection interval and slave latency.
  Use a connection supervision timeout of 900 ms.
  Use the common data channel selection parameters.
  The test uses the common current and next variables for SN and NESN.

![Diagram of LL/CON/MA/S/BV-14-C Test Procedure](image)

**Figure 4.239: LL/CON/MA/S/BV-14-C [Master Retransmission Request]**

1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.
2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters and sends a LL DATA packet with an invalid checksum in response to every packet, T_IFS after the packet from the IUT. Repeat 15 times.
3. During the repetition procedure, observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the next SN and by using the next NESN where the NESN matches the current SN in the previous packet correctly received.
4. Lower Tester expects the IUT to indicate in the following packets received correctly that the previous packet was not received correctly, with the SN matching the current NESN and the NESN matching the next SN.
5. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).
• Expected Outcome

Pass Verdict

The IUT transmits negative acknowledgements when sent a packet with an invalid CRC.

4.3.5.11 LL/CON/MAS/BV-15-C [Master Retransmission]

• Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme and retransmit a data packet on a negative acknowledgement.

The Lower Tester acts in the slave role, using negative acknowledgements prompts repeated retransmission requests from the IUT.

• Reference

[3] 4.5.9

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted).

• Test Procedure

1. Execute the test procedure using the common connection interval and slave latency.
2. Use a connection supervision timeout of 900 ms.
3. Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.

---

**Figure 4.240: LL/CON/MAS/BV-15-C [Master Retransmission]**

1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.
2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA in response to every packet, T_IFS after the packet received from the IUT, not acknowledge the packet received by
using the current NESN. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the current SN.

3. Lower Tester expects the IUT to retransmit the packet in the following packets received correctly, with the SN matching the current NESN and the NESN matching the next SN.

4. Repeat steps 2–3 15 times.

5. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

• Expected Outcome

   Pass Verdict

   The IUT retransmits the same payload when asked for a retransmission.

4.3.5.12 LL/CON/MAS/BV-16-C [Master Acknowledgement Repetition]

• Test Purpose

   Test that a master IUT is able to maintain a connection using the acknowledgement scheme and repeats a positive acknowledgement of a packet.

   The Lower Tester acts in the slave role, using negative acknowledgements to prompt a retransmission of an acknowledgement from the IUT.

• Reference

   [3] 4.5.9

• Initial Condition

   State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted).

• Test Procedure

   Execute the test procedure using the common connection interval and slave latency.

   Use a connection supervision timeout of 900 ms.

   Use the common data channel selection parameters.

   The test uses the common current and next variables for SN and NESN.
1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.

2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet in response to the IUT every packet, T_IFS after the packet received from the IUT. Not to recognize an acknowledgement, use the current SN where in the previous packet received the NESN is the next SN, but acknowledge packets correctly with the next NESN where the current NESN matches the SN in the previous packet correctly received.

3. Lower Tester expects the IUT to repeat the acknowledgement in the following packets received correctly, with the SN matching the current NESN and the NESN matching the next SN.

4. Repeat steps 2–3 15 times.

5. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- Expected Outcome

**Pass Verdict**

The IUT retransmits acknowledgements when negative acknowledgements are received.

### 4.3.5.13 LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]

- **Test Purpose**

Test that a master IUT is able to maintain a connection using the acknowledgement scheme and preserve the packet sequence numbering in the case of a lost negative acknowledgement.

The Lower Tester acts in the slave role, moving to the next packet after a negative acknowledgement to prompt a repeated negative acknowledgement from the IUT.
• Reference

[3] 4.5.9

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

• Test Procedure

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout of 900 ms.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.

Figure 4.242: LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]

1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data element with the value 0xFF with Packet_Boundary_Flag flag set.
2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet with a varying checksum to every packet, T_IFS after the packet from the IUT. To prompt the IUT to produce a negative acknowledgement, use the next SN for one packet to send with an invalid checksum, where the NESN in the previous packet received correctly is the next SN.
3. Lower Tester receives the IUT with negative acknowledgement and sends a DATA packet with a valid checksum but uses the SN equal to the SN in the previous Lower Tester packet. Acknowledge packets by using the next NESN where the current NESN matches the SN in the previous packet correctly received.
4. Lower Tester expects the IUT to continue the negative acknowledgement in the following packets received correctly, with the NESN matching the current SN.
5. Repeat step 3–4 15 times.
6. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- Expected Outcome

**Pass Verdict**

The IUT retransmits a negative acknowledgement when being transmitted the next payload.

### 4.3.5.14 LL/CON/MAS/BV-18-C [Master Latency Retransmission Request]

- **Test Purpose**

Test that a master IUT is able to maintain a connection using the acknowledgement scheme with the slave using latency.

The Lower Tester acts in the slave role, using slave latency and invalid checksums and observe the master handling the latency and retransmission requests from the IUT.

- **Reference**

[3] 4.5.9

- **Initial Condition**

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- **Test Procedure**

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout as defined by the TSPX_conn_timeout_max IXIT entry.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.
1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data element with the value 0xFF with Packet_Boundary_Flag flag set.

2. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet with an invalid checksum only in events required by the slave latency parameter, T_IFS after the packet received from the IUT. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the current SN and by using the next NESN where the current NESN matches the current SN in the previous packet correctly received.

3. Lower Tester expects the IUT to indicate in the following packets received correctly that the previous packet was not received correctly or no response was received, with the SN matching the current NESN and the NESN matching the current SN.

4. Repeat steps 2–3 15 times.

5. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- Expected Outcome

Pass Verdict

The IUT uses retransmissions to handle slave latency.
4.3.5.15  LL/CON/MAS/BV-19-C [Connection Control Timeout]

- **Test Purpose**
  Test that a master IUT terminates the link layer connection if a transaction is not completed before the connection control transaction timer expires.
  The Lower Tester acts in the slave role in the connection and avoids that a transaction is completed.

- **Reference**
  [3] 5.2

- **Initial Condition**
  Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- **Test Procedure**

  ![Test Procedure Diagram]

  **Figure 4.244: LL/CON/MAS/BV-19-C [Connection Control Timeout]**

  1. Upper Tester sends an HCI_LE_Read_REMOTE_FEATURE command and receives an HCI_Command_Status_Event.
  2. The IUT sends an LL_FEATURE_REQ PDU.
  3. Lower Tester acknowledges the LL_FEATURE_REQ PDU but does not send an LL_FEATURE_RSP PDU.
  4. IUT sends the event HCI_Disconnection_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester.

- **Expected Outcome**
  **Pass Verdict**
  The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires.
4.3.5.16  LL/CON/MAS/BV-20-C [Master Request Version]

- Test Purpose
  Test that a connected master IUT requests and performs the Version Exchange procedure.
  The Lower Tester acts in the slave role in a maintained connection and responds to the request from the IUT to perform version exchange.

- Reference
  [3] 5.1.5

- Initial Condition
  Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

1. Once connection is established, if the IUT has sent the Lower Tester a Version_Ind packet containing IUT version information by itself, the Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
2. If the IUT does not send the Lower Tester a Version_Ind packet containing IUT version information by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT and receives an HCI_Command_Status_Event as a response.
3. The IUT sends a Version_Ind packet containing IUT version information.
4. The Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
5. The IUT sends the event HCI_Read_Remote_Version_Information_Complete_Event with Tester version information to the Upper Tester.

Figure 4.245: LL/CON/MAS/BV-20-C [Master Request Version]
• Expected Outcome

Pass Verdict
The test procedure is executed successfully, with the IUT requesting the version information and acknowledging the reply.

If the procedure was initiated by the Upper Tester as described in step 2, the IUT reports the version requested completed with an HCI event.

4.3.5.17 LL/CON/MAS/BV-21-C [Master Respond Version]

• Test Purpose
Test that a connected master IUT responds to the request from the Lower Tester to perform the version exchange procedure.

The Lower Tester acts in the slave role in a maintained connection and requests to perform version exchange.

• Reference
[3] 5.1.5

• Initial Condition
Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• Test Procedure

1. Configure the Lower Tester to send Version_Ind packet.
2. Lower Tester sends a Version_Ind packet containing Tester version information.
3. The IUT acknowledges the Version_Ind packet and sends a Version_Ind packet containing IUT version information.

• Expected Outcome

Pass Verdict
The test procedure is executed successfully, with the IUT responding to the version information.

Inconclusive Verdict
The IUT sends an LL_VERSION_IND PDU before the Lower Tester does.
4.3.5.18 LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]

- Test Purpose
  Test that a master IUT is able to maintain a connection using the acknowledgement scheme.
  The Lower Tester acts in the slave role, maintaining the connection and checking that the IUT uses correctly the acknowledgement scheme.

- Reference
  [3] 4.5.9

- Initial Condition
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure
  Execute the test procedure using the common connection interval and slave latency. Use a connection supervision timeout of 900 ms. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN.

---

**Figure 4.247: LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]**

1. Configure Upper Tester to submit one data element to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0xFFFFFFF, for a data total length of 4, with Packet_Boundary_Flag flag set.
2. Configure Lower Tester to send one data packets with a payload length of 4 with the payload octet set to 0xFFFFFFF.
3. Lower Tester receives a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Expect the SN in a correctly received packet to match the current NESN and the NESN to match the next SN.
4. Lower Tester sends a DATA packet in response to every packet, T_IFS after the packet from the IUT. Observe the acknowledgement scheme by using the next SN for every packet to send where
the NESN in the packet correctly received is the next SN and by using the next NESN where the current NESN matches the SN in the packet correctly received.

5. The IUT sends the event HCI_LE_Data_Packet to the Upper Tester including the data sent by Lower Tester.
6. Repeat steps 1–5 15 times.
7. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- Expected Outcome

Pass Verdict
The test procedure executes successfully, with the IUT using the normal acknowledgement scheme operation.

4.3.5.19 LL/CON/MAS/BV-23-C [Responding to Feature Exchange]

- Test Purpose
Test that a connected master IUT responds to the feature exchange procedure and activates the correct features when requested.
The Lower Tester acts in the slave role in a maintained connection, transmits the request to perform feature exchange and observes the IUT responding.

- Reference
[8] 5.1.4.2

- Initial Condition
Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

![Diagram of LL/CON/MAS/BV-23-C](image)

*Figure 4.248: LL/CON/MAS/BV-23-C [Responding to Feature Exchange]*
1. Lower Tester sends an LL_SLAVE_FEATURE_REQ PDU including the Lower Tester’s feature set and waits for an LL_FEATURE_RSP PDU.
2. The IUT responds to the LL_SLAVE_FEATURE_REQ PDU with an LL_FEATURE_RSP PDU.

- Expected Outcome

  Pass Verdict
  All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.
  The test procedure is executed successfully, with the IUT responding with the feature response.

4.3.5.20 LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]

- Test Purpose

  Test that a master IUT is able to perform the connection parameter request procedure when the remote device accepts the request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and accepts the IUT’s request. The actual parameters used by the IUT may be different from the parameters provided by the Upper Tester.

- Reference

  [8] 5.1.7

- Initial Condition

  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• Test Procedure

Connection Established. IUT Master.

- **LL_CONNECTION_PARAM_REQ** (Min. Conn. Interval)
- **LL_CONNECTION_PARAM_RSP**
- **LL_CONNECTION_UPDATE_IND** (Min. Conn. Interval)
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **LL_CONNECTION_PARAM_REQ** (Max. Conn. Interval)
- **LL_CONNECTION_PARAM_RSP**
- **LL_CONNECTION_UPDATE_IND** (Max. Conn. Interval)
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **LL_CONNECTION_PARAM_REQ** (Min. Conn. Interval)
- **LL_CONNECTION_PARAM_RSP**
- **LL_CONNECTION_UPDATE_IND** (Min. Conn. Interval)
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **LL_CONNECTION_PARAM_REQ** (Max. Conn. Interval)
- **LL_CONNECTION_PARAM_RSP**
- **LL_CONNECTION_UPDATE_IND** (Max. Conn. Interval)
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**
- **Empty Data Packet**

**Figure 4.249: LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]**
Case 1:
1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester receives a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:
1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the maximum connection interval, no latency and maximum connection supervision timeout and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester receives a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:
1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT’s request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester receives a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

- For all three cases described in the test procedure, the following conditions shall occur:
  - The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
  - The IUT maintains the connection with the new parameters selected by the IUT in use.
  - The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.21 LL/CON/MAS/BV-25-C [Initiating Connection Parameter Request – Reject]

• Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device rejects the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT’s request.

• Reference

[8] 5.1.7

• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
**Test Procedure**

1. **Upper Tester** sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.

2. **Lower Tester** receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester rejects the IUT’s request by issuing an LL_REJECT_EXT_IND PDU containing ErrorCode 0x3B. Lower Tester receives a packet from the IUT acknowledging the LL_REJECT_EXT_IND PDU.

Alternative 1:
- Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the error code sent by the Lower Tester (0x3B).

Alternative 2:
- Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester).
- Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
- Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
- Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

  Pass Verdict

  The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.

  The IUT either reports a HCI_LE_Connection_Update_Complete event containing the error code sent by the Lower Tester (0x3B) or reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.22 LL/CON/MAS/BV-26-C [Initiating Connection Parameter Request – same procedure collision]

- Test Purpose

  Test that a master IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT’s connection parameter request and the remote device’s connection parameter request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new connection parameter request procedure upon receiving the IUT’s connection parameter request to cause a procedure collision. The test case expects the IUT to reject the slave’s connection parameter request and then proceed with its own connection parameter request procedure.

- Reference

  [8] 5.1.7

- Initial Condition

  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• **Test Procedure**

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_CONNECTION_PARAM_RSP.
3. Lower Tester expects the IUT to reject the Lower Tester's connection parameter request using an LL_REJECT_EXT_IND with reason code 0x23.
4. Lower Tester responds to the IUT’s connection parameter request using an LL_CONNECTION_PARAM_RSP and expects the IUT to respond with an LL_CONNECTION_UPDATE_IND.
5. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
6. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
7. At the time of the update start maintaining the connection with the new parameters.
8. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and rejects the Lower Tester’s request to update the connection parameters and then receives the connection parameter response from the Lower Tester.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.23 LL/CON/MAS/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

• Test Purpose

Test that a master IUT is able to perform the channel map update procedure when there is a procedure collision between the IUT’s channel map update and the Lower Tester’s connection parameter request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the channel map update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameter request procedure upon receiving the IUT’s channel map update to cause a procedure collision. The test case expects the IUT to reject the slave’s connection parameter request and then proceed with its own channel map update procedure.

• Reference

[8] 5.1.7

• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

1. Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT setting the channel map to only use even channels. Upper Tester receives an HCI_Command_Complete event from the IUT in response.
2. Lower Tester receives an LL_CHANNEL_MAP_IND control PDU from the IUT, with the parameters submitted in step 1.
3. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT with connection parameters set to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) as an acknowledgement packet to the LL_CHANNEL_MAP_IND PDU.
4. Lower Tester expects the IUT to reject the LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
5. Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the channel map update request.

• Expected Outcome

Pass Verdict

The IUT transmits the LL_CHANNEL_MAP_IND PDU to update the channel map and rejects the Lower Tester’s connection parameter request.

The IUT maintains the connection with the Lower Tester with the updated data channel selection parameters after the assigned event.
4.3.5.24  LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

- **Test Purpose**
  Test that a master IUT is able to perform the encryption start procedure when there is a procedure collision between the IUT’s encryption start procedure and the Lower Tester’s connection parameter request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the encryption procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameter request procedure upon receiving the IUT’s encryption start to cause a procedure collision. The test case expects the IUT and Lower Tester to complete the encryption start procedure followed by completing the connection parameter request procedure.

- **Reference**
  [8] 5.1.7, 5.1.3.1

- **Initial Condition**
  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
- **Test Procedure**

  ![Flowchart](Diagram.png)

  **Figure 4.253: LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]**

  1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT. Receive an HCI_Command_Status event from the IUT in response.
  2. Lower Tester receives an LL_ENC_REQ PDU from the IUT.
  3. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT with connection parameters set to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) as an acknowledgement packet to the LL_ENC_REQ PDU.
  4. Lower Tester responds with an LL_ENC_RSP PDU.
  5. Lower Tester sends an LL_START_ENC_REQ to the IUT and expects the IUT to acknowledge it.
  6. Lower Tester receives an LL_START_ENC_RSP from the IUT and responds with an LL_START_ENC_RSP.
  7. The Upper Tester receives an HCI_Encryption_Change event from the IUT.
  8. At any time after step 3, and possibly interlaced with steps 4 to 7, the IUT requests the Upper Tester to accept or reject the Lower Tester’s request to update the connection parameters. The Upper Tester accepts the request.
9. Lower Tester expects the IUT to send an LL_CONNECTION_UPDATE_IND (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester).
10. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
11. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
12. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- **Expected Outcome**

  **Pass Verdict**
  The IUT sends the LL_ENC_REQ PDU.
  The IUT sends an LL_START_ENC_RSP PDU until acknowledged.
  The IUT successfully reports the encryption change with the HCI event HCI_Encryption_Change.
  The IUT responds positively to the Lower Tester’s request to update connection parameters.
  The IUT maintains the connection with the new parameters selected by the IUT in use.
  The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

- **Notes**

  The Lower Tester and Upper Tester ensure that the encryption start procedure completes before the procedure response timeout for the connection parameter request procedure fires.

### 4.3.5.25 LL/CON/MAS/BV-29-C [Initiating Connection Parameter Request – remote legacy host]

- **Test Purpose**
  Test that a master IUT is able to perform the connection parameter request procedure when the remote device’s host either does not support the connection parameter request procedure or has masked the remote connection parameters request event.
  
The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT’s request.

- **Reference**

  [8] 5.1.7

- **Initial Condition**

  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX
  
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• Test Procedure

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester rejects the IUT's request by issuing an LL_REJECT_EXT_IND PDU containing ErrorCode 0x1A. Lower Tester receives a packet from the IUT acknowledging the LL_REJECT_EXT_IND PDU.
3. Lower Tester receives an LL_CONNECTION_UPDATE_IND control PDU from the IUT.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.

• Expected Outcome

Pass Verdict
- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
- The IUT sends an LL_CONNECTION_UPDATE_IND PDU to the Lower Tester when the Lower Tester sends an LL_REJECT_EXT_IND PDU with ErrorCode 0x1A in response to the IUT’s LL_CONNECTION_PARAM_REQ PDU.
- The IUT maintains the connection with the new parameters selected by the IUT in use.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

### 4.3.5.26 LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred_Periodicity]

- **Test Purpose**
  Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave does not indicate any preferred periodicity.

- **Reference**
  [8] 5.1.7

- **Initial Condition**
  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• Test Procedure

Connection Established. IUT Master.

LL CONNECTION PARAM REQ
(Min. Conn_INTERVAL)

HCI LE Remote_Connection_Parameter_Request
HCI LE Remote_Connection_Parameter_Request
HCI Command_Complete_Event
(Status: 0x00)

LL CONNECTION UPDATE IND
(Min. Conn_INTERVAL)

Empty Data Packet
Empty Data Packet
Empty Data Packet

LL CONNECTION PARAM REQ
(Max. Conn_INTERVAL)

HCI LE Remote_Connection_Parameter_Request
HCI LE Remote_Connection_Parameter_Request
HCI Command_Complete_Event
(Status: 0x00)

LL CONNECTION UPDATE IND
(Max. Conn_INTERVAL)

Empty Data Packet
Empty Data Packet
Empty Data Packet

LL CONNECTION PARAM REQ
(Min. Conn_INTERVAL)

HCI LE Remote_Connection_Parameter_Request
HCI LE Remote_Connection_Parameter_Request
HCI Command_Complete_Event
(Status: 0x00)

LL CONNECTION UPDATE IND
(Min. Conn_INTERVAL)

Empty Data Packet
Empty Data Packet
Empty Data Packet

Figure 4.255: LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred_Periodicity]
Case 1:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the maximum connection interval, no latency, and maximum connection supervision timeout and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester accepts the request.
3. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

For all the three cases described in the test procedure, the following conditions shall occur:

- The IUT responds positively to the Lower Tester’s request to update connection parameters.
- The IUT maintains the connection with the new parameters selected by the IUT in use.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.27  LL/CON/MAS/BV-31-C [Accepting Connection Parameter Request – preferred anchor points only]

• Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave only requests a change in anchor points.

• Reference

[8] 5.1.7

• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
Case 1:
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are 1.25ms from the old anchor points. Lower Tester acknowledges the connection update request.

3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.

4. Once the event count matches the time, the new parameters such as the new connection interval will be used.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to (connection interval – 1.25ms), Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.

2. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are (connection interval - 1.25ms) from the old anchor points. Lower Tester acknowledges the connection update request.

3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.

4. Once the event count matches the time, the new parameters such as the new connection interval will be used.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to 2.5ms, Offset2-5 to invalid and the connection interval, latency and supervision timeout unchanged.

2. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are 1.25ms from the old anchor points. Lower Tester acknowledges the connection update request.

3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.

4. Once the event count matches the time, the new parameters such as the new connection interval will be used.

• Expected Outcome

Pass Verdict
- The IUT responds positively to the Lower Tester’s request to update connection parameters.
- The IUT maintains the connection with the new parameters in use:
  - In the first case, the connection events are shifted by 1.25ms.
  - In the second case, the connection events are shifted by (connection interval – 1.25ms).
  - In the third case, the connection events are shifted by 1.25ms.

4.3.5.28 [Accepting Connection Parameter Request]

• Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave indicates parameters as specified in Table 4.56.

• Reference

[8] 5.1.7
• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester
address, supported type of own address, connection interval, common slave latency, common
timeout).

• Test Procedure

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection
parameters to a connection interval (with a non-zero range), no latency, intermediate connection
supervision timeout (3 s) and a preferred periodicity such that there is at least one connection
interval that is a multiple of the preferred periodicity as specified in Table 4.56.

2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper
Tester accepts the request.

3. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT. Lower Tester
acknowledges the connection update request.

4. Maintain the connection using empty DATA packets until the event count matches the time
indicated in the connection update request packet.

5. Once the event count matches the time, the new parameters such as the new connection interval
selected by the IUT will be used.

6. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the
new connection parameters.
• **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Slave Parameter Request</th>
<th>Preferred Periodicity</th>
<th>Offset0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.3.5.28.1 LL/CON/MAS/BV-32-C</strong> [Accepting Connection Parameter Request – Preferred_Periodicity]</td>
<td>Preferred Periodicity</td>
<td>Preferred periodicity within the connection interval range</td>
<td>0</td>
</tr>
<tr>
<td><strong>4.3.5.28.2 LL/CON/MAS/BV-33-C</strong> [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points]</td>
<td>Preferred Periodicity and preferred anchor points</td>
<td>Preferred periodicity within the connection interval range, a reference connection event counter and a valid Offset0 value such that the new connection event is 1.25ms away from the old connection event at the reference connection event count (Offset1-5 are invalid)</td>
<td>Valid</td>
</tr>
</tbody>
</table>

*Table 4.56: Accepting Connection Parameter Request Test Cases*

• **Expected Outcome**

**Pass Verdict**

The IUT responds positively to the Lower Tester’s request to update connection parameters.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

**4.3.5.29 LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked]**

• **Test Purpose**

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave requires the master LL to request for approval from the master’s Host and the master’s Host has masked the LE Remote Connection Parameter Request Event.

• **Reference**

[8] 5.1.7

• **Initial Condition**

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• Test Procedure

![Diagram of the test procedure](image)

**Figure 4.258: LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked]**

1. Upper Tester masks the LE Remote Connection Parameter Request event on the IUT.
2. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
3. Lower Tester receives an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1A.

• Expected Outcome

**Pass Verdict**

The IUT responds to the Lower Tester's request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode (0x1A).

4.3.5.30 LL/CON/MAS/BV-35-C [Accepting Connection Parameter Request – Host rejects]

• Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the master’s Host rejects the slave’s connection parameter request procedure.

• Reference

[8] 5.1.7

• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
**Test Procedure**

![Diagram: Link Layer (LL) Test Procedure](image)

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
2. The IUT requests the Upper Tester to accept or reject the Lower Tester’s request. The Upper Tester rejects the request using error code 0x3B.
3. Lower Tester receives an LL_REJECT_EXT_IND control PDU from the IUT containing the ErrorCode provided by the Upper Tester.

**Expected Outcome**

**Pass Verdict**

The IUT responds to the Lower Tester’s request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode provided by the Upper Tester.

**4.3.5.31 LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]**

**Test Purpose**

Test that a master IUT is able to perform the PHY update procedure. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT and accepts the IUT’s request.

**Reference**

[10] 5.1.10

**Initial Condition**

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• **Test Procedure**

![Diagram of test procedure](image)

*Figure 4.260: LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]*

The following steps shall be carried out 2N times as follows, where N is the number of cases in [Table 4.57](#) (selected based on the supported PHY(s)):

1. **Upper Tester** sends an HCI_LE_Set_PHY command to the IUT with the payload defined in the HCI_LE_Set_PHY section of [Table 4.57](#) and PHY_options set to 0x0000.
2. The Upper Tester receives an HCI_Command_Status event from the IUT in response. If any bits set in TX_PHYS or RX_PHYS correspond to unsupported PHYs, the Status shall be set to "Unsupported Feature or Parameter Value (0x11)". Otherwise the Status shall be set to zero.
3. If the IUT does not initiate a PHY change, proceed to step 9 if the Status in step 2 was set to zero or proceed to the next round if the Status in step 2 was set to a non-zero value.
4. The Lower Tester receives an LL_PHY_REQ control PDU from the IUT with at least one bit set in each field (RX_PHYS, TX_PHYS). The Lower Tester acknowledges the IUT’s request and responds with an LL_PHY_RSP PDU with the payload defined in the LL_PHY_RSP section of [Table 4.57](#).
5. Lower Tester receives an LL_PHY_UPDATE_IND with zero or one bits set in each field (M_TO_S_PHY, S_TO_M_PHY) and a selected PHY present in the payload sent in the LL_PHY_RSP PDU. If no bits are set in either field, proceed to step 8.
6. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
7. Once the event count matches the time, the new PHY(s) selected by the IUT will be used.
8. IUT sends empty DATA packets to the Lower Tester, and the Lower Tester acknowledges these packets, using the selected PHY(s).

9. If the command was accepted in step 2 or at least one of the PHY fields in the LL_PHY_UPDATE_IND PDU was non-zero, the Upper Tester receives an LE_PHY_Update_Complete event from the IUT with a payload consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in LL_PHY_UPDATE_IND was zero or LL_PHY_UPDATE_IND was not sent). Otherwise the Upper Tester receives no event.

<table>
<thead>
<tr>
<th>Case</th>
<th>HCI_LE_Set_PHY</th>
<th>LL_PHY_RSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>1</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>3</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>4</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>5</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>6</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>7</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>8</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>9</td>
<td>0x00</td>
<td>0x03</td>
</tr>
<tr>
<td>10</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>11</td>
<td>0x02</td>
<td>0x02</td>
</tr>
<tr>
<td>12</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>13</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>14</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>15</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>16</td>
<td>0x00</td>
<td>0x01</td>
</tr>
<tr>
<td>17</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>18</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>19</td>
<td>0x00</td>
<td>0x01</td>
</tr>
</tbody>
</table>
### Table 4.57: PDU payload contents for each case variation

<table>
<thead>
<tr>
<th>Case</th>
<th>HCI_LE_Set_PHY</th>
<th>LL_PHY_RSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALL_PHYS</td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>20</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>21</td>
<td>0x00</td>
<td>0x05</td>
</tr>
<tr>
<td>22</td>
<td>0x01</td>
<td>0x00</td>
</tr>
<tr>
<td>23</td>
<td>0x02</td>
<td>0x04</td>
</tr>
<tr>
<td>24</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>25</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>26</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>27</td>
<td>0x00</td>
<td>0x06</td>
</tr>
<tr>
<td>28</td>
<td>0x00</td>
<td>0x02</td>
</tr>
<tr>
<td>29</td>
<td>0x00</td>
<td>0x04</td>
</tr>
<tr>
<td>30</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>31</td>
<td>0x03</td>
<td>0x00</td>
</tr>
<tr>
<td>32</td>
<td>0x03</td>
<td>0x00</td>
</tr>
</tbody>
</table>

### Expected Outcome

**Pass Verdict**

For all cases described in the test procedure, the following conditions shall occur:

- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in both the LL_PHY_REQ and LL_PHY_RSP PDUs, or zero in the following cases:
  
  a. The LL_PHY_REQ and LL_PHY_RSP have no common PHY for that field.
  b. The LL_PHY_REQ and LL_PHY_RSP both specify the current PHY (no change).
  c. The LL_PHY_UPDATE_IND has a zero for both fields.

- If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.

- The IUT maintains the connection with the PHY(s) selected by the LL_PHY_UPDATE_IND, only changing PHY(s) if LL_PHY_UPDATE_IND PDU was sent and a change was indicated in the PDU.

- The IUT reports the selected PHY(s) with a LE_PHY_Update_Complete event, even if the PHY(s) did not change, if the command was accepted in step 2. The contents of the
LE_PHY_Update_Complete event are consistent with the LL_PHY_UPDATE_IND if the PHY changed or the prior PHY if no change occurred.

- The IUT does not send an LE_PHY_Update_Complete event if the command was rejected in step 2 and either the IUT did not initiate the PHY Update Procedure or it initiated the procedure but no PHY change occurred.

**Fail Verdict**
The IUT accepts the command in step 2 when a bit set in TX_PHYS or RX_PHYS corresponds to an unsupported PHY.

**Inconclusive Verdict**
The IUT does not initiate at least one PHY Update Procedure during this test case.

### 4.3.5.32 LL/CON/MAS/BV-42-C [Initiating PHY Update Procedure – Symmetric Only]

- **Test Purpose**
  Test that a master IUT is able to perform the PHY update procedure when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT and accepts the IUT’s request.

- **Reference**
  [10] 5.1.10

- **Initial Condition**
  Same as LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure].

- **Test Procedure**
  Same as LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure] except that, in step 1, when ALL_PHYS is 0x00 and TX_PHYS does not equal RX_PHYS, the IUT shall return the error code Unsupported Feature or Parameter Value (0x11) to the Upper Tester (it may still initiate a PHY change).

- **Expected Outcome**
  **Pass Verdict**
  Same as LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]. In addition:
  - Each time the LL_PHY_REQ PDU is sent by the IUT, it shall have the same value in the TX_PHYS and RX_PHYS fields and this value shall have exactly one bit set.
  - Each time the LL_PHY_UPDATE_IND PDU is sent by the IUT, it shall have the same value in the M_TO_S_PHY and S_TO_M_PHY fields.

  **Inconclusive Verdict**
  The IUT does not initiate at least one PHY Update Procedure during this test case.
4.3.5.33  LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure]

- **Test Purpose**
  Test that a master IUT is able to respond to a PHY update procedure from a slave device. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

- **Reference**
  [10] 5.1.10

- **Initial Condition**
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- **Test Procedure**

  ![Diagram illustrating the test procedure](image)

  **Figure 4.261: LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure]**

1. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the ALL_PHYS field set to a value of 0x03. The Upper Tester receives an HCI_Command_Status event indicating success in response. The controller may send a LL_PHY_REQ to the Lower Tester. In this case, the Lower Tester sends a LL_PHY_RSP specifying the current PHY in both directions in
response and the IUT completes the transaction with an LL_PHY_UPDATE_IND. Whether or not the procedure is carried out with the Lower Tester, the Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

2. Perform steps 3 through 9 2N times as follows, where N is the number of cases in Table 4.58, Table 4.59, or Table 4.60 (selected based on the supported PHY(s)):
   - firstly using cases 1 to N from the relevant table in order;
   - then using the cases from the relevant table in a random order.

3. Lower Tester sends an LL_PHY_REQ PDU to the IUT with the payload specified in the relevant table.

4. Lower Tester receives an LL_PHY_UPDATE_IND PDU from the IUT with a value selected for M_TO_S_PHY and S_TO_M_PHY that is either a bit value present in the LL_PHY_REQ or zero, with a maximum of 1 bit set for each field. If either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.

5. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.

6. Once the event count matches the time, the PHY(s) selected by the IUT in the LL_PHY_UPDATE_IND packet will be used.

7. At the Instant of the PHY change start maintaining the connection with the selected PHY(s).

8. IUT sends empty DATA packets to the Lower Tester, and Lower Tester acknowledges these packets, using the selected PHY(s).

9. If the PHY(s) were changed, Upper Tester receives an LE_PHY_Update_Complete event from the IUT containing the PHY(s) selected. If both PHYs were NOT changed, Upper Tester does NOT receive an LE_PHY_Update_Complete event.

<table>
<thead>
<tr>
<th>Case</th>
<th>LL_PHY_REQ</th>
<th>TX_PHYS</th>
<th>RX_PHYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0x01</td>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0x02</td>
<td>0x01</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0x01</td>
<td>0x01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0x03</td>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0x03</td>
<td>0x01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0x01</td>
<td>0x03</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0x02</td>
<td>0x03</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0x03</td>
<td>0x03</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.58: PDU payload contents for each case variation for LE 2M PHY.
### Table 4.59: PDU payload contents for each case variation for LE Coded PHY

<table>
<thead>
<tr>
<th>Case</th>
<th>LL_PHY_REQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>1</td>
<td>0x04</td>
</tr>
<tr>
<td>2</td>
<td>0x01</td>
</tr>
<tr>
<td>3</td>
<td>0x04</td>
</tr>
<tr>
<td>4</td>
<td>0x01</td>
</tr>
<tr>
<td>5</td>
<td>0x05</td>
</tr>
<tr>
<td>6</td>
<td>0x05</td>
</tr>
<tr>
<td>7</td>
<td>0x01</td>
</tr>
<tr>
<td>8</td>
<td>0x04</td>
</tr>
<tr>
<td>9</td>
<td>0x05</td>
</tr>
<tr>
<td>10</td>
<td>0x04</td>
</tr>
<tr>
<td>11</td>
<td>0x05</td>
</tr>
<tr>
<td>12</td>
<td>0x05</td>
</tr>
<tr>
<td>Case</td>
<td>LL_PHY_REQ</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>13</td>
<td>0x04</td>
</tr>
<tr>
<td>14</td>
<td>0x03</td>
</tr>
<tr>
<td>15</td>
<td>0x06</td>
</tr>
<tr>
<td>16</td>
<td>0x03</td>
</tr>
<tr>
<td>17</td>
<td>0x01</td>
</tr>
<tr>
<td>18</td>
<td>0x05</td>
</tr>
<tr>
<td>19</td>
<td>0x04</td>
</tr>
<tr>
<td>20</td>
<td>0x01</td>
</tr>
<tr>
<td>21</td>
<td>0x05</td>
</tr>
<tr>
<td>22</td>
<td>0x01</td>
</tr>
<tr>
<td>23</td>
<td>0x01</td>
</tr>
<tr>
<td>24</td>
<td>0x03</td>
</tr>
<tr>
<td>25</td>
<td>0x06</td>
</tr>
<tr>
<td>26</td>
<td>0x02</td>
</tr>
<tr>
<td>27</td>
<td>0x06</td>
</tr>
<tr>
<td>28</td>
<td>0x02</td>
</tr>
<tr>
<td>29</td>
<td>0x03</td>
</tr>
<tr>
<td>30</td>
<td>0x07</td>
</tr>
<tr>
<td>31</td>
<td>0x02</td>
</tr>
<tr>
<td>32</td>
<td>0x03</td>
</tr>
<tr>
<td>33</td>
<td>0x02</td>
</tr>
<tr>
<td>34</td>
<td>0x04</td>
</tr>
<tr>
<td>35</td>
<td>0x07</td>
</tr>
<tr>
<td>36</td>
<td>0x07</td>
</tr>
<tr>
<td>Case</td>
<td>LL_PHY_REQ</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>TX_PHYS</td>
</tr>
<tr>
<td>37</td>
<td>0x06</td>
</tr>
<tr>
<td>38</td>
<td>0x02</td>
</tr>
<tr>
<td>39</td>
<td>0x07</td>
</tr>
<tr>
<td>40</td>
<td>0x04</td>
</tr>
<tr>
<td>41</td>
<td>0x02</td>
</tr>
<tr>
<td>42</td>
<td>0x06</td>
</tr>
<tr>
<td>43</td>
<td>0x07</td>
</tr>
<tr>
<td>44</td>
<td>0x07</td>
</tr>
<tr>
<td>45</td>
<td>0x06</td>
</tr>
<tr>
<td>46</td>
<td>0x06</td>
</tr>
<tr>
<td>47</td>
<td>0x05</td>
</tr>
<tr>
<td>48</td>
<td>0x07</td>
</tr>
<tr>
<td>49</td>
<td>0x01</td>
</tr>
</tbody>
</table>

Table 4.60: PDU payload contents for each case variation for LE 2M and LE Coded PHY.

- **Expected Outcome**

**Pass Verdict**

For all cases described in the test procedure, the following conditions shall occur:

- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in the LL_PHY_REQ, or zero. In addition, if the two fields of the LL_PHY_REQ held the same value which only had a single bit set, then the M_TO_S_PHY and S_TO_M_PHY fields shall either both contain that value or shall both be zero.

- If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.

- The IUT maintains the connection with the PHY(s) selected by the LL_PHY_UPDATE_IND payload.

- If the PHY(s) were changed, IUT sends a LE_PHY_Update_Complete event containing the PHY(s) selected in the LL_PHY_UPDATE_IND payload. If both PHYs were NOT changed, IUT does NOT send a LE_PHY_Update_Complete event.
Inconclusive Verdict
The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.5.34 LL/CON/MAS/BV-44-C [Responding to PHY Update Procedure – Symmetric Only]

- Test Purpose
  Test that a master IUT is able to respond to a PHY update procedure from a slave device when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

- Reference
  [10] 5.1.10

- Initial Condition
  Same as LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure].

- Test Procedure
  Same as LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure].

- Expected Outcome
  Pass Verdict
  Same as LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure]. In addition:
  - Each time the LL_PHY_UPDATE_IND PDU is sent by the IUT, it shall have the same value in the M_TO_S_PHY and S_TO_M_PHY fields.

Inconclusive Verdict
The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.5.35 LL/CON/MAS/BV-45-C [Handling Protocol Collision – Same Procedure]

- Test Purpose
  Test that a master IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT’s PHY change request and the remote device’s PHY change request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY change request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT’s PHY change request to cause a procedure collision. The test case expects the IUT to reject the slave’s PHY change request and then proceed with its own PHY change request procedure.

- Reference
  [10] 5.3

- Initial Condition
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. Lower Tester receives an LL_PHY_REQ control PDU from the IUT with at least one bit set in each field (RX_PHYS, TX_PHYS). If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.

3. Lower Tester responds with an LL_PHY_REQ with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M.

4. Lower Tester expects the IUT to reject the Lower Tester's LL_PHY_REQ using an LL_REJECT_EXT_IND with reason code 0x23.

5. Lower Tester responds to the IUT's LL_PHY_REQ using an LL_PHY_RSP PDU with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M and expects the IUT to respond with an LL_PHY_UPDATE_IND with a value selected for M_TO_S_PHY and S_TO_M_PHY that is either a bit value present in the LL_PHY_REQ or zero, with a maximum of 1 bit set for each field. If either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value. If both fields are zero the test case ends with an Inconclusive Verdict.

6. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.

7. Once the event count matches the time, the PHY(s) selected by the IUT in the LL_PHY_UPDATE_IND will be used (or prior PHY(s) if no change specified).
8. At the instant of the PHY change start maintaining the connection with the selected PHY(s).
9. IUT sends empty DATA packets to the Lower Tester, and Lower Tester acknowledges these packets, using the selected PHY(s).
10. Upper Tester receives an LE_PHY_Update_Complete from the IUT with values corresponding to the PHY(s) selected in the LL_PHY_UPDATE_IND PDU, or the prior PHY(s) if no change specified.

• Expected Outcome

Pass Verdict
- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in the LL_PHY_RSP, or zero in the following cases:
  a. The LL_PHY_REQ and LL_PHY_RSP have no common PHY for that field.
  b. The LL_PHY_REQ and LL_PHY_RSP both specify the current PHY (no change).
  c. The LL_PHY_RSP had the same single PHY selected for both directions and the LL_PHYUPDATE_IND has a zero for both fields.
- If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the instant shall have a valid value.
- The IUT reports the selected PHY(s) with a LE_PHY_Update_Complete event to the host, even if no change occurs.

Inconclusive Verdict
The IUT does not initiate the PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2, or does not select a PHY change in step 5.

4.3.5.36 LL/CON/MAS/BV-46-C [Protocol Timeout for PHY Update Procedure]

• Test Purpose
Test that a master IUT terminates the Link Layer connection if the master-initiated PHY update procedure is not completed before the procedure response timer expires.

The Lower Tester acts in the slave role in the connection and ensures that the procedure initiated by the IUT is not completed.

• Reference
[10] 5.2

• Initial Condition
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• **Test Procedure**

![Diagram showing test procedure](image)

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with both fields (TX_PHYS, RX_PHYS) set to prefer a PHY other than LE 1M and PHY_options set to 0x0000 and receives an HCI_Command_Status event.
2. The IUT sends an LL_PHY_REQ PDU with at least one bit set of each field (RX_PHYS, TX_PHYS). If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
3. Lower Tester acknowledges the LL_PHY_REQ PDU but does not send an LL_PHY_RSP PDU.
4. The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero status.
5. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

• **Expected Outcome**

**Pass Verdict**

The IUT sends HCI_Disconnect_Complete event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in step 4 the status shall be non-zero.

**Inconclusive Verdict**

The IUT does not initiate the PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).
4.3.5.37 LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]

- **Test Purpose**
  Test that a master IUT is able to perform the channel map update procedure when there is a procedure collision between the IUT’s channel map update and the Lower Tester’s PHY change request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the channel map update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT’s channel map update to cause a procedure collision. The test case expects the IUT to reject the slave’s PHY change request and then proceed with its own channel map update procedure.

- **Reference**
  [10] 5.3

- **Initial Condition**
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- **Test Procedure**

  ![Diagram of Test Procedure](image)

  *Figure 4.264: LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]*
1. Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT setting the channel map to only use even channels. Upper Tester receives an HCI_Command_Complete event from the IUT in response.

2. Lower Tester receives an LL_CHANNEL_MAP_IND control PDU from the IUT, with the parameters submitted in step 1.

3. Lower Tester responds with an LL_PHY_REQ with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M.

4. Lower Tester expects the IUT to reject the Lower Tester’s LL PHY_REQ using an LL_REJECT_EXT_IND with reason code 0x2A.

5. Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the channel map update request.

   • Expected Outcome

   Pass Verdict

   The IUT transmits the LL_CHANNEL_MAP_IND PDU to update the channel map and rejects the Lower Tester’s request to update the selected PHY(s).

   The IUT maintains the connection with the Lower Tester with the updated data channel selection parameters after the assigned event.

4.3.5.38 LL/CON/MAS/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

   • Test Purpose

   Test that a master IUT is able to perform the connection parameters request procedure when there is a procedure collision between the IUT’s connection parameters request and the Lower Tester’s PHY change request.

   The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameters request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT’s connection parameters request to cause a procedure collision. The test case expects the IUT to reject the slave’s PHY change request and then proceed with its own connection parameters request procedure.

   • Reference

   [10] 5.3

   • Initial Condition

   State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
Test Procedure

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3s) and receives an HCI_Command_Status event from the IUT in response.
2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_PHY_REQ with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M.
3. Lower Tester expects the IUT to reject the Lower Tester’s LL_PHY_REQ using an LL_REJECT_EXT_IND with reason code 0x2A.
4. Lower Tester responds to the IUT’s connection parameters request using an LL_CONNECTION_PARAM_RSP and expects the IUT to respond with an LL_CONNECTION_UPDATE_IND.
5. Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the connection update request packet.
6. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
7. At the time of the update start maintaining the connection with the new parameters.
8. Upper Tester receives an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
• Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and rejects the Lower Tester’s request to update the selected PHY(s).

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.39  LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]

• Test Purpose

Test that a master IUT follows all packet time restrictions both during and after PHY change when it initiates the PHY update procedure.

The Lower Tester acts in the slave role maintaining a connection. A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT and issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester observes the PHY update procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the procedure.

• Reference

[10] 5.1.10.1

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
Test Procedure

Connection Established. IUT Master

Lower Tester 
IUT 
Upper Tester

---

Figure 4.266: LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 2M PHY and receives an HCI_Command_Status event from the IUT in response.
4. The Lower Tester receives an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_PSP with the RX_PHYS and TX_PHYS fields both set to 0x02.
5. Lower Tester receives an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE 2M PHY for both directions. If the IUT does not select the LE 2M PHY for both directions the test case ends with an Inconclusive Verdict.

6. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.

7. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and receives an HCI_Command_Complete event from the IUT in response.

8. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT’s TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.

9. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

10. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

11. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHY and TX_PHY both set to prefer the LE 1M PHY.

12. The Lower Tester receives an LL_PHY_REQ from the IUT and responds with an LL_PHY_RSP with the RX_PHY and TX_PHY fields both set to 0x01.

13. The Lower Tester receives an LL_PHY_UPDATE_IND from the IUT selecting the LE 1M PHY to be used in both directions. If the IUT does not select the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.

14. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 13 must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY update procedure.

15. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

16. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.

17. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

18. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

• Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in step 13) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_LL_PHY_UPDATE_IND in step 13) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.
All data packets received by the Lower Tester shall:
- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

**Inconclusive Verdict**

One or more of the following:
- The IUT does not select the LE 1M PHY for both directions in step 2).
- The IUT does not select the LE 2M PHY for both directions in step 5).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in step 8).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in step 13).
- The IUT does not select the LE 1M PHY for both directions in step 13).

**4.3.5.40 LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions]**

- **Test Purpose**
  Test that a master IUT both during and after PHY change follows all packet time restrictions when it responds to a PHY update procedure from a slave device.

The Lower Tester acts in the slave role maintaining a connection. A PHY update procedure is performed to set both directions to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queueing data to the IUT and starts the PHY update procedure. The Lower Tester observes the PHY update procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the change.

- **Reference**
  [10] 5.1.10.1

- **Initial Condition**
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

Connection Established. IUT Master

Lower Tester initiates Data Length Procedure

IUT Initiates Data Length Procedure

UT queues data packets for remainder of the test case

Figure 4.268: LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 2M PHY and receives an HCI_Command_Status event from the IUT in response.

2. The Lower Tester receives an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_RSP with the RX_PHYS and TX_PHYS fields both set to 0x02.

Figure 4.269: LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions - Part B]
3. Lower Tester receives an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE 2M PHY for both directions. If the IUT does not select the LE 2M PHY for both directions the test case ends with an Inconclusive Verdict.
4. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
5. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and receives an HCI_Command_Complete event from the IUT in response.
6. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and receive a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
7. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
8. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
9. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and receives an HCI_Command_Status event from the IUT in response. The controller may send a LL_PHY_REQ to the Lower Tester. In this case, the Lower Tester sends an LL_PHY_RSP specifying the current PHY in both directions in response and the IUT completes the transaction with an LL_PHY_UPDATE_IND.
10. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 2M PHY.
11. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x01.
12. Lower Tester receives an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions. If the IUT does not select the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.
13. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 12 must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY update procedure.
14. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
15. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.
16. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
17. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in step 12) must have a length of 32 octets and be un-fragmented.
All data packets received by the Lower Tester after the IUT sends an LL_PHY_UPDATE_IND in step 12) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:
- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:
- The IUT does not select the LE 2M PHY for both directions in step 3).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in step 6).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in step 12).
- The IUT does not select the LE 1M PHY for both directions in step 12).

4.3.5.41 LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]

- Test Purpose

Test that a master IUT correctly handles the case where it initiates a PHY update procedure but no common PHYS are available.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT.

- Reference

[10] 5.1.10

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• **Test Procedure**

![Diagram of test procedure](image)

**Figure 4.270: LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]**

1. Upper Tester sends an HCI\_LE\_Set\_PHY command to the IUT with ALL\_PHYS set to zero, RX\_PHYS and TX\_PHYS both set to prefer a single supported PHY other than LE 1M, and PHY\_options set to 0x0000 and receives an HCI\_Command\_Status event from the IUT in response.

2. The Lower Tester receives an LL\_PHY\_REQ control PDU from the IUT with at least one bit set in each field (RX\_PHYS, TX\_PHYS). If the IUT does not send an LL\_PHY\_REQ the test case ends with an Inconclusive Verdict.

3. Lower Tester acknowledges the IUT’s request and responds with an LL\_PHY\_RSP PDU. For each field the Lower Tester will select a value based on the following rules:
   a. If the IUT sets a single bit in a field of the LL\_PHY\_REQ, the Lower Tester will set a different bit in the corresponding field of the LL\_PHY\_RSP. The bit set must correspond to a PHY that the IUT supports.
   b. If the IUT sets more than one bit in a field of the LL\_PHY\_REQ, the Lower Tester will set the bit corresponding to the current PHY.

4. Lower Tester receives an LL\_PHY\_UPDATE\_IND with both the M\_TO\_S\_PHY and S\_TO\_M\_PHY fields set to zero. The Instant field must also be zero.

5. Upper Tester receives an LE\_PHY\_Update\_Complete event from the IUT with a payload consistent with the prior PHY(s) indicating no change has occurred.

• **Expected Outcome**

**Pass Verdict**

For all cases described in the test procedure, the following conditions shall occur:

- The IUT transmits an LL\_PHY\_UPDATE\_IND PDU where the M\_TO\_S\_PHY and S\_TO\_M\_PHY fields are both set to zero (no change). The Instant field is also set to zero.
- The IUT reports the prior PHY(s) with a LE\_PHY\_Update\_Complete event indicating no change has occurred.
Inconclusive Verdict

The IUT does not send an LL_PHY_REQ in step 2).

4.3.5.42  LL/CON/MAS/BV-52-C [Master Receiving Data, LE Coded, CI Change]

- **Test Purpose**

  Test that a master IUT is able to receive data from a slave device when the slave is transitioning between 125kbit and 500kbit coded rates. Confirm that IUT responds within the allowed T_IFS times for each packet at either coded rate. Test is performed with the IUT’s minimum and maximum supported packet length. A Data Length Update Procedure is performed if required.

  The Lower Tester acts in the slave role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- **Reference**

  [10] 4.5

- **Initial Condition**

  Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, maximum supported connection interval, common slave latency, common timeout).

- **Test Procedure**

  Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT ALL_PHYS set to zero, RX_PHYS and TX_PHYS both set to prefer LE Coded PHY, and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. The Lower Tester receives an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_RSP with the RX_PHYS and TX_PHYS fields both set to prefer the LE Coded PHY.

3. Lower Tester receives an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
4. The Upper Tester receives an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE Coded PHY.
5. If (TSPX_TxOctets_Max > 27) OR (TSPX_TxTime_Max > 328) then the Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to TSPX_TxTime_Max and TxOctets set to TSPX_TxOctets_Max and receives an HCI_Command_Complete event from the IUT in response, otherwise go to step 7.
6. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max and receives a response from the IUT.
7. Configure Lower Tester to send 288 data packets of length connEffectiveMaxTxOctets with all payload octets set to continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.). For each packet sent the CI field will alternate in the pattern below. Each number in the pattern represents the CI field for that particular packet to be sent.

```
2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 8 2 2 2 2 2 2 2 2
2 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 2 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 2 8 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
2 8 8 8 8 8 8 8 2 8 2 2 2 2 2 2
```
8. Lower Tester sends one or more DATA packets to the IUT, using the acknowledgement scheme and the data channel selection parameters. Lower Tester receives an empty DATA packet in response from the IUT for each DATA packet sent by the Lower Tester, received with acceptable T_IFS timing (150 µs +/- 2 µs).
9. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing one or more data elements sent in step 8. The received data should be in the form of continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.). Repeat steps 7–9 until all data to be sent has been reported to the Upper Tester.
10. Repeat the entire data exchange procedure again from steps 7–10, but utilizing the minimum data payload size (27 bytes) instead of connEffectiveMaxTxOctets for each data packet.

- Expected Outcome

  **Pass Verdict**

  The test procedure completes with the IUT acknowledging all the data sent.

  The IUT reports all data correctly with HCI_Data_Packet events containing continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 … 0xFE, 0xFF, 0x00, 0x01, etc.).

  All responses from the IUT are received with acceptable T_IFS timing (150 µs +/- 2 µs).

4.3.5.43  **LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]**

- Test Purpose

  Test that a master IUT follows all packet time restrictions both during and after PHY update when it initiates the PHY Update Procedure. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

  The Lower Tester acts in the slave role maintaining a connection. A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT and issues the HCI command to start the PHY Update Procedure as the Host of the IUT. The Lower Tester observes the PHY Update Procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the procedure.

- Reference

  [10] 5.1.10.1

- Initial Condition

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

Connection Established. IUT Master

IUT Initiates Data Length Procedure

Lower Tester initiates Data Length Procedure

UT queues data packets for remainder of the test case

Figure 4.272: LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]
Figure 4.273: LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.
4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure,
the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.

5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.

7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHYS and TX_PHYS both set to prefer LE Coded PHY.

8. The Lower Tester receives an LL_PHY_REQ from the IUT and responds with an LL_PHY_RSP with the RX_PHYS and TX_PHYS fields both set to 0x04.

9. The Lower Tester receives an LL_PHY_UPDATE_IND from the IUT selecting the LE Coded PHY to be used in both directions. If the IUT does not select the LE Coded PHY for both directions, the test case ends with an Inconclusive Verdict.

10. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 9 must be 140 octets or less in length. If a larger packet is received, the test case ends with a failed verdict. The Lower Tester receives data packets both before and after the PHY Update Procedure.

11. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

12. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.

13. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.

14. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

- Expected Outcome

**Pass Verdict**

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in step 9 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_ LL_PHY_UPDATE_IND in step 9 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.
Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 1M PHY for both directions in step 2.
- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in step 9.
- The IUT does not select the LE Coded PHY for both directions in step 9.
- The IUT sends autonomously an LL_LENGTH_REQ after the instant specified in the LL_PHY_UPDATE_IND in step 9.

4.3.5.44 LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded]

- Test Purpose
  Test that a master IUT both during and after PHY update when it responds to a PHY Update Procedure from a slave device. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

  The Lower Tester acts in the slave role maintaining a connection. A Data Length Update Procedure is performed. The Upper Tester begins queueing data to the IUT and starts the PHY Update Procedure. The Lower Tester observes the PHY Update Procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the change.

- Reference
  [10] 5.1.10.1

- Initial Condition
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

Connection Established. IUT Master

IUT Initiates Data Length Procedure

Lower Tester initiates Data Length Procedure

OR

Empty Data Packet

Data Packet

UT queues data packets for remainder of the test case

Continued in Part B...

Figure 4.274: LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LECoded – Part A]
1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and receives an HCI_Command_Status event from the IUT in response.

2. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and receives an HCI_Command_Complete event from the IUT in response.

4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and
RxOctets set to 141 and receive a response from the IUT. If the IUT’s TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.

5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

6. The Lower Tester receives un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with a fail verdict.

7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.

8. Lower Tester receives an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 13 must be 140 octets or less in length. If a larger packet is received the test case ends with a fail verdict. The Lower Tester receives data packets both before and after the PHY Update Procedure.

10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.

11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.

12. The Upper Tester receives an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.

13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.

• Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in step 8 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_ LL_PHY_UPDATE_IND in step 8 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 1M PHY for both directions in step 2.
- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in step 8.
The IUT does not select the LE Coded PHY for both directions in step 8.

4.3.5.45 LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded]

• Test Purpose

Test that a master IUT still transmits data even when the TxTime and/or RxTime values for LE Coded PHY suggest a smaller possible data length than the minimum length data PDU (27 octets).

The Lower Tester acts in the slave role maintaining a connection. If a Data Length Update Procedure is performed, minimum settings are used. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure. Lower Tester observes the procedure carried out by the IUT and ensures PDUs are still transmitted after the procedure.

• Reference

[10] 5.1.9

• Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• Test Procedure

Figure 4.276: LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded – Part A]
The test procedure and Expected Outcome are the same as test LL/CON/SLA/BV-57-C except that step 6 is:

6. Lower Tester receives an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.

---

**Figure 4.277: LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded - Part B]**

The test procedure and Expected Outcome are the same as test LL/CON/SLA/BV-57-C except that step 6 is:

6. Lower Tester receives an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
4.3.5.46 LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]

- **Test Purpose**
  Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE 1M PHY.

- **Reference**
  [8] 5.1.9, [10] 4.5.10

- **Initial Condition**
  State: Connected Master. Values for maximum TxOctets supported (supportedMaxTxOctets_{IUT}), TxTime (supportedMaxTxTime_{IUT}), RxOctets (supportedMaxRxOctets_{IUT}) and RxTime (supportedMaxRxTime_{IUT}) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

  Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:
  - connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
  - connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in “Octet”
  - connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in “Time”
  - connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in “Time”

  The values of these parameters will depend on the features supported by the IUT.
- **Test Procedure**

![Diagram of test procedure](image)

**For each round from 1 to 16**

1. **LL_LENGTH_REQ**
2. **LL_LENGTH_RSP**
   - (RxOctets, RxTime, TxOctets, TxTime)

   - If RxOctets, RxTime, TxOctets, or TxTime changes:
     - **HCI_LE_Data_Length_Change**
     - (RxOctets, RxTime, TxOctets, TxTime)

3. One or more **HCI_ACL_Data_Packet**
   - (Data_Total_Length > TxOctets,
     Data_Octets > TxOctets)

Repeat until data is sent

   - **Data Packet**
   - (Data_Length <= TxOctets,
     Packet Tx Time <= Txtime)

Repeat for at least 20 Packets

   - **Data Packet**
   - (Data_Length <= min(connEffectiveMaxRxOctets,
     TxOctets))

Repeat until data is sent

   - **HCI_ACL_Data_Packet**
   - (Data_length)

---

1. **For each round in** [Table 4.61](#), **perform steps 2–9**. **Note**: if two or more rounds turn out to use the same values, all except the first may be omitted.

2. **The Lower Tester sends an LL_LENGTH_REQ to the IUT with the payload specified for the relevant round.**

3. **The Lower Tester receives an LL_LENGTH_RSP from the IUT with a value selected for:**
   - connMaxTxOctetsIUT between connMinOctetsLimitSpec and supportedMaxTxOctetsIUT
   - connMaxTxTimeIUT between connMinTimeLimitSpec and supportedMaxTxTimeIUT
   - connMaxRxOctetsIUT between connMinOctetsLimitSpec and supportedMaxRxOctetsIUT
   - connMaxRxTimeIUT between connMinTimeLimitSpec and supportedMaxRxTimeIUT

4. **If the values in either the LL_LENGTH_REQ or LL_LENGTH_RSP PDUs mean the values of** connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime have changed, the **Upper Tester receives an HCI_LE_Data_Length_Change event** from the IUT containing the new values. Otherwise it does not receive such an event. **Note**: in the very first round it might not be possible to determine whether the values have changed.
5. The Upper Tester sends the data to the IUT in the minimum possible number of HCI ACL DATA packets based on the TxOctets. If more than one packet is used, they shall all be sent together immediately after one connection event closes and before the next one opens; connection interval shall be make large enough to make that possible.

6. The Lower Tester receives DATA packets until all the data has been transmitted, with each packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no longer than connEffectiveMaxTxTime microseconds to transmit.

7. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester in step 5.

8. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to smaller of the IUT’s connEffectiveMaxRxOctets and the maximum number of octets that can be included in a packet that can be transmitted on the current PHY in no longer than the IUT’s connEffectiveMaxRxTime microseconds.

9. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects the data transmitted by the IUT to match the data sent by the Lower Tester in step 8.

10. If the IUT supports the LE Coded PHY, then repeat steps 2–9 but only performing the rounds where either connMaxTxTimeTester or connMaxRxTimeTester is connMaxTimeLimitSpec and, in those rounds, replacing connMaxTimeLimitSpec by 2120 µs.

<table>
<thead>
<tr>
<th>Round</th>
<th>LL_LENGTH_REQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>connMaxTxOctetsTester</td>
</tr>
<tr>
<td>1</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>2</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>3</td>
<td>supportedMaxRxOctetsIUT</td>
</tr>
<tr>
<td>4</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>5</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>6</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>7</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>8</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>9</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>10</td>
<td>supportedMaxRxOctetsIUT</td>
</tr>
<tr>
<td>11</td>
<td>supportedMaxRxOctetsIUT</td>
</tr>
<tr>
<td>12</td>
<td>supportedMaxRxOctetsIUT</td>
</tr>
<tr>
<td>13–16</td>
<td>supportedMaxRxOctetsIUT</td>
</tr>
</tbody>
</table>

Table 4.61: LL_LENGTH_REQ content
• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT responding to the Lower Tester’s LL_LENGTH_REQ PDU with an LL_LENGTH_RSP PDU.

The IUT sends an HCI_LE_Data_Length_Change event if at least one of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime has changed since the previous round and the event contains the correct values. The IUT does not send an event if none of the values have changed.

The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations.

4.3.5.47 LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]

• Test Purpose

Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE 1M PHY.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the Data Length update procedure as the Host of the IUT, and the Lower Tester then observes the Data Length update procedure carried out by the IUT and accepts the IUT’s request.

• Reference

[8] 5.1.9
[10] [8] 4.5.10

State: Connected Master. Values for maximum TxOctets supported (supportedMaxTxOctets \_IUT), TxTime (supportedMaxTxTime \_IUT), RxOctets (supportedMaxRxOctets \_IUT) and RxTime (supportedMaxRxTime \_IUT) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in “Octet”
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in “Time”
- connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in “Time”

The values of these parameters will depend on the features supported by the IUT.
Test Procedure

For each round from 1 to 12

1. For each round in Table 4.62, perform steps 2–10. Note: if two or more rounds turn out to use the same values, all except the first may be omitted.

2. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with the payload specified for the relevant round.

3. The Lower Tester receives an LL_LENGTH_REQ from the IUT with a value selected for:
   - \( \text{connMaxTxOctets}_{\text{IUT}} \) between \( \text{connMinOctetsLimitSpec} \) and \( \text{supportedMaxTxOctets}_{\text{IUT}} \)
   - \( \text{connMaxTxTime}_{\text{IUT}} \) between \( \text{connMinTimeLimitSpec} \) and \( \text{supportedMaxTxTime}_{\text{IUT}} \)
   - \( \text{connMaxRxOctets}_{\text{IUT}} \) between \( \text{connMinOctetsLimitSpec} \) and \( \text{supportedMaxRxOctets}_{\text{IUT}} \)
   - \( \text{connMaxRxTime}_{\text{IUT}} \) between \( \text{connMinTimeLimitSpec} \) and \( \text{supportedMaxRxTime}_{\text{IUT}} \)

   If no such PDU is sent, skip to the start of the next round.

4. The Lower Tester responds with an LL_LENGTH_RSP PDU with the two octet fields set to \( \text{connMinOctetsLimitSpec} \) and the two time fields set to \( \text{connMinTimeLimitSpec} \).

5. If the values in either the LL_LENGTH_REQ or LL_LENGTH_RSP PDUs mean the values of \( \text{connEffectiveMaxTxOctets} \), \( \text{connEffectiveMaxRxOctets} \), \( \text{connEffectiveMaxTxTime} \), or
connEffectiveMaxRxTime have changed, the Upper Tester receives an 
HCI_LE_Data_Length_Change event from the IUT containing the new values. Otherwise it does 
not receive such an event. Note: in the very first round it might not be possible to determine 
whether the values have changed. Note: the IUT might send two events if it makes an 
autonomous change and then the Lower Tester’s response also causes an event.

6. The Upper Tester sends the data to the IUT in the minimum possible number of HCI ACL DATA 
packets based on the TxOctets. If more than one packet is used, they shall all be sent together 
immediately after one connection event closes and before the next one opens; connection 
interval shall be make large enough to make that possible.

7. The Lower Tester receives DATA packets until all the data has been transmitted, with each 
packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no 
longer than connEffectiveMaxTxTime microseconds to transmit.

8. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester 
in step 6.

9. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to 
smaller of the IUT’s connEffectiveMaxRxOctets and the maximum number of octets that can be 
included in a packet that can be transmitted on the current PHY in no longer than the IUT’s 
connEffectiveMaxRxTime microseconds.

10. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects 
the data transmitted by the IUT to match the data sent by the Lower Tester in step 9.

11. Repeat steps 1–10 but, in step 4, set the two octet fields to connMaxOctetsLimitSpec and the two 
time fields to connMaxTimeLimitSpec.

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Data_Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TxOctets</td>
</tr>
<tr>
<td>1</td>
<td>supportedMaxTxOctetsIUT</td>
</tr>
<tr>
<td>2</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>3</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>4</td>
<td>supportedMaxTxOctetsIUT</td>
</tr>
<tr>
<td>5</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>6</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>7</td>
<td>supportedMaxTxOctetsIUT</td>
</tr>
<tr>
<td>8</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>9</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>10</td>
<td>supportedMaxTxOctetsIUT</td>
</tr>
<tr>
<td>11</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>12</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td>13</td>
<td>supportedMaxTxOctetsIUT</td>
</tr>
</tbody>
</table>
### Table 4.62: HCI_LE_Set_Data_Length content

<table>
<thead>
<tr>
<th>Round</th>
<th>HCI_LE_Set_Data_Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td></td>
<td>TxOctets</td>
</tr>
<tr>
<td></td>
<td>2128 µs</td>
</tr>
<tr>
<td>15</td>
<td>connMaxOctetsLimitSpec</td>
</tr>
<tr>
<td></td>
<td>TxTime</td>
</tr>
<tr>
<td></td>
<td>2128 µs</td>
</tr>
</tbody>
</table>

**Expected Outcome**

**Pass Verdict**

The test procedure is executed successfully, with the IUT transmitting an LL_LENGTH_REQ PDU to the Lower Tester.

The IUT sends an HCI_LE_Data_Length_Change event if at least one of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime has changed since the previous round and the event contains the correct values. The IUT does not send an event if none of the values have changed.

The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations.

**Inconclusive Verdict**

The IUT does not initiate at least one Data Length Update Procedure by sending an LL_LENGTH_REQ to the Lower Tester during this test case.

### 4.3.5.48 LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY]

**Test Purpose**

Verify that the IUT as Master correctly handles communication with a Lower Tester that does not support the Data Length Update Procedure.

**Reference**

[8] 5.1.9

**Initial Condition**

State: Connected Master. Values for maximum TxOctets supported (supportedMaxTxOctets IUT), TxTime (supportedMaxTxTime IUT), RxOctets (supportedMaxRxOctets IUT) and RxTime (supportedMaxRxTime IUT) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in “Octet”
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in “Time”
- connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in "Time"

The values of these parameters will depend on the features supported by the IUT.

- **Test Procedure**

---

**Figure 4.280: LL/CON/MAS/BV-75-C: Master Data Length Update – Responding to Data Length Update Procedure**

1. Repeat step 2 until the Lower Tester receives an LL_LENGTH_REQ from the PDU or until it has been repeated 20 times. In the latter case, return an Inconclusive Verdict.
2. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT.
   a. The first time, the TxOctets field shall be set to supportedMaxTxOctets\textsubscript{IUT} and the TxTime field to supportedMaxTxTime\textsubscript{IUT}. 

b. The second time, the TxOctets field shall be set to the lesser of \( \text{connMinOctetsLimitSpec} + 1 \) and \( \text{supportedMaxTxOctets}_\text{IUT} \) and the TxTime field to the lesser of \( \text{connMinTimeLimitSpec} + 1 \) and \( \text{supportedMaxTxTime}_\text{IUT} \).

c. The remaining times, the TxOctets field shall be set to a random value between \( \text{connMinOctetsLimitSpec} \) and \( \text{supportedMaxTxOctets}_\text{IUT} \) inclusive and the TxTime field to a random value between \( \text{connMinTimeLimitSpec} \) and \( \text{supportedMaxTxTime}_\text{IUT} \) inclusive.

3. The Lower Tester checks that the LL_LENGTH_REQ PDU received from the IUT has a value selected for
   - \( \text{connMaxTxOctets}_\text{IUT} \) between \( \text{connMinOctetsLimitSpec} \) and \( \text{supportedMaxTxOctets}_\text{IUT} \)
   - \( \text{connMaxTxTime}_\text{IUT} \) between \( \text{connMinTimeLimitSpec} \) and \( \text{supportedMaxTxTime}_\text{IUT} \)
   - \( \text{connMaxRxOctets}_\text{IUT} \) between \( \text{connMinOctetsLimitSpec} \) and \( \text{supportedMaxRxOctets}_\text{IUT} \)
   - \( \text{connMaxRxTime}_\text{IUT} \) between \( \text{connMinTimeLimitSpec} \) and \( \text{supportedMaxRxTime}_\text{IUT} \)

4. The Lower Tester responds with an LL_UNKNOWN_RSP PDU with the Unknown type set to LL_LENGTH_REQ.

5. The Upper Tester sends the data to the IUT in the minimum possible number of HCI ACL DATA packets based on the TxOctets. If more than one packet is used, they shall all be sent together immediately after one connection event closes and before the next one opens; connection interval shall be make large enough to make that possible.

6. The Lower Tester receives DATA packets until all the data has been transmitted, with each packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no longer than connEffectiveMaxTxTime microseconds to transmit.

7. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester in step 5.

8. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to smaller of the IUT’s connEffectiveMaxRxOctets and the maximum number of octets that can be included in a packet that can be transmitted on the current PHY in no longer than the IUT’s connEffectiveMaxRxTime microseconds.

9. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects the data transmitted by the IUT to match the data sent by the Lower Tester in step 8.

- Expected Outcome

  **Pass Verdict**

  The test procedure is executed successfully, with the IUT transmitting an LL_LENGTH_REQ PDU to the Lower Tester.

  The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations.

  **Fail Verdict**

  The IUT sends an HCI_LE_DATA_LENGTH_CHANGE event to the Upper Tester.

  **Inconclusive Verdict**

  The IUT does not initiate at least one Data Length Update Procedure by sending an LL_LENGTH_REQ to the Lower Tester during this test case.
4.3.5.49 LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]

- **Test Purpose**
  
  Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE 2M PHY.

- **Test Procedure**

  ![Diagram]

  **Figure 4.281**: LL/CON/MAS/BV-76-C: Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY

  This test is identical to LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY] except that, before step 1, execute this following step:

  - The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x02 (LE 2M PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an
HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE 2M PHY, return an Inconclusive Verdict.

- Expected Outcome

The same as LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY] and add the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE 2M PHY.

4.3.5.50 LL/CON/MAS/BV-77-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]

- Test Purpose

Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE 2M PHY.
Test Procedure

Connection Established. IUT as Master.

For each round from 1 to 12

Repeat until data is sent

Data Packet

Repeat for at least 20 Packets

Data Packet

Repeat until data is sent

HCI_ACL_Data_Packet

Figure 4.282: LL/CON/MAS/BV77-C: Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY

This test is identical to LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY] except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x02 (LE 2M PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE 2M PHY, return an Inconclusive Verdict.
• Expected Outcome
The same as LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY] and add in the following Inconclusive Verdict:

Inconclusive Verdict
The IUT does not change to the LE 2M PHY.

4.3.5.51 LL/CON/MAS/BV-78-C [Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY]

• Test Purpose
Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE Coded PHY.

• Test Procedure

Figure 4.283: LL/CON/MAS/BV-78-C: Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY
This test is identical to LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY] except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x04 (LE Coded PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.

- Expected Outcome

  The same as LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY] and add in the following Inconclusive Verdict:

  **Inconclusive Verdict**

  The IUT does not change to the LE Coded PHY.

4.3.5.52 LL/CON/MAS/BV-79-C [Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY]

- Test Purpose

  Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE Coded PHY.
**Test Procedure**

![Test Diagram]

- **Connection Established. IUT as Master.**

  - **LL_PHY_REQ**
  - **LL_PHY_RSP**
  - **LL_PHY_UPDATE_IND**

  - **HCI_LE_PHY_Update_Complete**

  - For each round from 1 to 12
    - **HCI_LE_Set_Data_Length**
      - 
        - **(RxOctets, RxTime, TxOctets, TxTime)**
        - **LL LENGTH_RSP**
        - **(RxOctets, RxTime, TxOctets, TxTime)**

    - **If RxOctets, RxTime, TxOctets, or TxTime changes**
      - **HCI_LE_Data_Length_Change**
        - **(RxOctets, RxTime, TxOctets, TxTime)**
    
    - **One or more**
      - **HCI_ACL_Data_Packet**
        - **(Data_Total_Length > TxOctets, Data_Octets > TxOctets)**

    - **Repeat until data is sent**
      - **Data Packet**
        - **(Data_Length <= TxOctets, Packet Tx Time <= TxTime)**

    - **Repeat for at least 20 Packets**
      - **Data Packet**
        - **(Data_Length:min(connEffectiveMaxRxOctets, TxOctets))**

    - **Repeat until data is sent**
      - **HCI_ACL_Data_Packet**
        - **(Data_Length)**

**Figure 4.284: LL/CON/MAS/BV-79-C: Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY**

This test is identical to LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY] except that, before step 1, execute this following step:

- The Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x04 (LE Coded PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an **HCI_LE_PHY_Update_Complete** event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.
• Expected Outcome

The same as LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY] and add in the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE Coded PHY.

4.3.5.53  LL/CON/MAS/BV-80-C [Master Data Length Update – Slave does not support; LE Coded PHY]

• Test Purpose

Verify that the IUT as Master correctly handles communication with a Lower Tester that does not support the Data Length Update Procedure on LE Coded PHY.
• Test Procedure

This test is identical to LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY] except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYs and RX_PHYs set to 0x04 (LE Coded PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an
HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.

- **Expected Outcome**

  The same as LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY] and add in the following Inconclusive Verdict:

  **Inconclusive Verdict**

  The IUT does not change to the LE Coded PHY.

**4.3.5.54 LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]**

- **Test Purpose**

  Test that a master IUT is able to perform the connection parameter request procedure when a feature exchange has not been performed and the remote device does not support the request.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and sends an LL_UNKNOWN_RSP PDU in response to the IUT’s request.

- **Reference**

  [8] 5.1.7

- **Initial Condition**

  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
• Test Procedure

![Diagram](image)

**Figure 4.286: LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]**

1. If the IUT autonomously initiates a feature exchange before step 3, the test ends with an Inconclusive Verdict.
2. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
3. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LLUNKNOWN_RSP PDU with the Opcode field set to LL_CONNECTION_PARAM_REQ (0x0F).
4. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester).
5. Maintain the connection using empty data packets until the event count matches the time indicated in the connection update request packet.
6. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
7. Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.
8. Repeat steps 2 and 4-7, using connection parameter values in step 2 different than in the first iteration. If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.
• Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters, then it sends an LL_CONNECTION_UPDATE_IND PDU.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

In the second iteration, the IUT skips the LL_CONNECTION_PARAM_REQ PDU and sends an LL_CONNECTION_UPDATE_IND PDU.

4.3.5.55 LL/CON/MAS/BV-82-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

• Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure after the feature exchange reveals that the remote device does not support the request.

The Lower Tester acts in the slave role maintaining a connection and initiates feature exchange, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT.

• Reference

[8] 5.1.7

• Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).
Test Procedure

1. Lower Tester initiates a feature exchange, unless the IUT has already done so. Lower Tester indicates that it does not support the Connection Parameter Request Procedure.
2. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.
3. Lower Tester receives an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). If the IUT sends an LL_CONNECTION_PARAM_REQ PDU instead of the LL_CONNECTION_UPDATE_IND PDU, the test ends with a Fail Verdict.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. Upper Tester receives an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.

Expected Outcome

Pass Verdict
The IUT transmits an LL_CONNECTION_UPDATE_IND PDU.
The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.
4.3.5.56 LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]

- **Test Purpose**
  Test that a master IUT does not change the PHY when it requests an asymmetrical PHY connection, but the slave indicates it only supports a symmetrical connection.

  The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT.

- **Reference**
  [10] 5.1.10

- **Initial Condition**
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout). All communications shall use the 1M PHY.

- **Test Procedure**

  ![Diagram of test procedure](image)

  **Figure 4.288: LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]**

  1. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x00, TX_PHY set to 0x01, and RX_PHY set to 0x02.
  2. The Upper Tester receives an HCI_Command_Status event from the IUT in response.
  3. The IUT initiates a PHY change by issuing an LL_PHY_REQ command to the Lower Tester with TX_PHY set to 0x01 and RX_PHY set to 0x02. If the IUT does not send an LL_PHY_REQ PDU or sends LL_PHY_REQ PDU with different values in TX_PHY or RX_PHY, the test case ends with an Inconclusive Verdict.
  4. The Lower Tester responds with LL_PHY_RSP with TX_PHY set to 0x02 and TX_PHY set to 0x02.
5. The IUT sends an LL_PHY_UPDATE_IND to the Lower Tester with M_TO_S_PHY set to 0x00 and S_TO_M_PHY set to 0x00.
6. The Upper Tester receives an LE_PHY_Update_Complete event from the IUT with a payload indicating no change.

- Expected Outcome

**Pass Verdict**
The IUT completes the sequence by issuing an LL_PHY_UPDATE_IND with M_TO_S_PHY set to 0x00 and S_TO_M_PHY set to 0x00, indicating no change.

**Inconclusive Verdict**
The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU or sends sending an LL_PHY_REQ PDU with TX_PHY not set to 0x01 and RX_PHY not set to 0x02 in step 3).

4.3.5.57 **LL/CON/MAS/BI-02-C [Master T_Terminate Timer]**

- **Test Purpose**
Test the correct behavior of a master device when TERMINATE_IND packets are not acknowledged.

The Lower Tester acts in the slave role, receives TERMINATE_IND packets from the IUT and does not acknowledge them, until T_Terminate expires.

- **Reference**
[3] 5.1.6

- **Initial Condition**
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- **Test Procedure**
Execute the test procedure using the common connection parameters.

---

Figure 4.289: LL/CON/MAS/BI-02-C [Master T_Terminate Timer]
1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps’ execution and receives an HCI_Command_Status in response.

2. Lower Tester expects the IUT to transmit a TERMINATE_IND packet. Lower Tester does not acknowledge the termination packet, but continues the slave transmissions.

3. Lower Tester expects the retransmission of TERMINATE_IND packet from the IUT and responds with an empty DATA packet not acknowledging the termination packet. Repeat until T_Terminate expires.

4. Upper Tester receives an HCI_Disconnection_Complete event including status of 0x00 (success), and a reason code of 0x16 ("Connection Terminated by Local Host") or 0x22 ("LL Response Timeout") from the IUT indicating loss of the link.

   • Expected Outcome

   Pass Verdict

   The IUT reports the connection termination with an HCI event.

   The IUT keeps sending TERMINATE_IND packets until T_Terminate timer expires.

4.3.5.58  LL/CON/MAS/BI-04-C [Master Connection Control Timer]

   • Test Purpose

   Test that a slave device is able to recover from a control procedure failure.

   The Lower Tester acts in the slave role, failing to acknowledge a control procedure request in order to expire the master connection control timer.

   • Reference

   [3] 5.2

   • Initial Condition

   State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

   • Test Procedure

   Execute the test procedure using the common connection parameters.
1. Once connection is established, if the IUT has sent LL_VERSION_IND packet by itself, Lower Tester acknowledges the version packet. Lower Tester continues the slave transmissions but it never sends the LL_VERSION_IND packet. Then, continue on step 4.

2. If the IUT does not send LL_VERSION_IND packet by itself, Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT containing the connection handle from the preamble steps' execution and receives an HCI_Command_Status in response.

3. Lower Tester expects the IUT to transmit a VERSION_IND packet. Lower Tester acknowledges the version packet.

4. Lower Tester continues sending empty DATA packet until the connection control timeout value, or until step 5 executes.

5. Upper Tester receives an HCI_Disconnection_Complete event from the IUT indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from LL response timeout.
• Expected Outcome

Pass Verdict

The test procedure executes successfully with the IUT using the connection control timer to recover from the failed control procedure.

4.3.5.59 LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]

• Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device does not respond to the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and does not respond to the IUT’s request.

• Reference

[3] 5.1.7

• Initial Conditions

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

• Test Procedure

![Diagram of LL/CON/MAS/BI-05-C Test Procedure]

Figure 4.291: LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]
1. Upper Tester sends an HCI_LE_Connectio_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and receives an HCI_Command_Status event from the IUT in response.

2. Lower Tester receives an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). The Lower Tester acknowledges the LL_CONNECTION_PARAM_REQ PDU but does not respond to the LL_CONNECTION_PARAM_REQ.

3. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

• Expected Outcome

  **Pass Verdict**
  
  The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

4.3.5.60 LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]

• Test Purpose

  Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave contains illegal parameters.

• Reference

  [3] 5.1.7

• Initial Conditions

  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

• Test Procedure

![Diagram](image-url)

*Figure 4.292: LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]*
1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection interval min and max to outside the valid range i.e., 4 (6ms).
2. Lower Tester receives an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1E.

• Expected Outcome
  
  Pass Verdict
  
The IUT responds to the Lower Tester’s request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode.

4.3.6 Both Connected Roles

4.3.6.1 Constant Tone Extension Request Procedure, IUT Responding, Unsupported

• Test Purpose
  
  Tests that the IUT responds correctly to a Constant Tone Extension Request Procedure initiated by the Lower Tester when the IUT does not support the proper CTE type or length. Test that the IUT generates an LL_REJECT_EXT_IND PDU.
  
The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT for each parameter and confirms the IUT correctly sends an LL_REJECT_EXT_IND PDU.

• Test Case IDs
  
  LL/CON/SLA/BV-87-C
  LL/CON/MAS/BV-83-C

• Reference
  
  [10] 5.1.12

• Initial Condition
  
  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
  
The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.

• Test Procedure
  
  Execute the test procedure for IUT as Master (if supported) and IUT as Slave (if supported).
1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. If the IUT supports AoD, the Upper Tester expects the IUT to also return a Max_Length_Switching_Pattern between 0x02 and 0x4B. The Upper Tester stores the Max_CTE_Length and the Max_Length_Switching_Pattern (if applicable) for future use.

2. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Transmit_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. CTE_Types shall be set to a supported value. If AoD is selected, Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDS[0] through Antenna_IDS[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, …, TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDS[] values.

3. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).

4. If the Max_CTE_Length value is equal to 0x14, skip to step 7.

5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to Max_CTE_Length + 1. CTETypeReq is set to a value that matches the CTE_Types value specified in step 2.

6. The Lower Tester receives an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

7. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to 0x02. CTETypeReq is set to a value that does not match the CTE_Types value specified in step 2.

8. The Lower Tester receives an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).
• Expected Outcome

Pass Verdict

The IUT sends an LL_REJECT_EXT_IND PDU with RejectOpCode set to LL_CTE_REQ and ErrorCode set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

4.3.6.2  [Periodic Advertising Sync Transfer Procedure, Advertising IUT Initiated]

• Test Purpose

Tests that the IUT with periodic advertising enabled is able to initiate the Periodic Advertising Sync Transfer Procedure. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well. The test purpose is exercised over each supported PHY, including using different PHYs for the connection and periodic advertising. The case that the remote does not support the procedure is tested.

The Lower Tester verifies that the IUT continues periodic advertising and that the IUT transfers synchronization information that is valid for that periodic advertising. The Upper Tester commands the IUT to perform the procedure and observes the responses returned by the IUT.

• Reference

[13] 4.6.23

• Initial Condition

MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

• Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in Table 4.63 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
<th>Primary ADV PHY</th>
<th>Secondary ADV PHY</th>
<th>Test Privacy and Unknown Rsp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Step 1</td>
<td>Step 2</td>
<td>Step 2, Step 10</td>
<td>Step 15</td>
</tr>
</tbody>
</table>

| LL/CON/SLA/BV-88-C | LE 1M | 0x01 (LE 1M) | 0x01 (LE 1M) | Yes   |
| LL/CON/MAS/BV-84-C |

| LL/CON/SLA/BV-89-C | LE 1M | 0x03 (LE Coded) | 0x03 (LE Coded) | No    |
| LL/CON/MAS/BV-85-C |

| LL/CON/SLA/BV-90-C | LE 1M | 0x01 (LE 1M) | 0x02 (LE 2M) | No    |
| LL/CON/MAS/BV-86-C |

| LL/CON/SLA/BV-91-C | LE Coded | 0x01 (LE 1M) | 0x01 (LE 1M) | No    |
| LL/CON/MAS/BV-87-C |

| LL/CON/SLA/BV-92-C | LE 2M | 0x01 (LE 1M) | 0x01 (LE 1M) | No    |
| LL/CON/MAS/BV-88-C |

Table 4.63: Values for each Test Case ID
1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.63, and the Lower Tester connects with the IUT on that PHY.

2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT. Advertising_Event_Properties parameter is set to 0x0000. Own_Address_Type is set to 0x00 (Public Address). The Primary_Advertising_PHY is set to the value indicated in the Primary ADV PHY column in Table 4.63. The Secondary_Advertising_PHY is set to the value indicated in the Secondary ADV PHY column in Table 4.63.

3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT. Periodic_Advertising_Properties parameter is set to 0x0000.
4. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

5. The Upper Tester enables advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command.

6. The Lower Tester receives an ADV_EXT_IND PDU from the IUT with the AuxPtr Extended Header field present.

7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the SyncInfo Extended Header fields present.

8. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements.

9. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Set_Info_Transfer command to the IUT with the Advertising_Handle of the advertising configured in step 2, Service_Data set to any non-zero value, and the Connection_Handle of the connection and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

10. The Lower Tester receives an LL_PERIODIC_SYNC_IND PDU from the IUT with ID set to the value used for Service_Data in step 9, SyncInfo and connEventCount set to valid values, SID set to match the advertising set ID representing the advertising in step 2, AType set to the address type used for the advertising in step 2, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in Table 4.63, and the AdvA set to the IUT’s advertiser address used for the advertising in step 2.

11. The Lower Tester verifies that the IUT continues maintaining the connection.

12. The Lower Tester uses the SyncInfo and connEventCount values to verify that they allow synchronization to the periodic advertising as in step 8.

13. The Upper Tester disables extended advertising using the HCI_LE_Set_EXTENDED_Advertising_Enable command.

14. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.

15. If the Test Privacy and Unknown Rsp column in Table 4.63 indicates No, then skip to step 18.

16. If the IUT supports controller-based privacy, then repeat steps 2–14, except that in step 2 the Own_Address_Type is set to 0x02 (Resolvable Private Address) and Peer_Address_Type and Peer_Address are set to values matching the Lower Tester’s address matching a local IRK in the resolving list, and that in step 10, the AdvA is set to the IUT’s unresolved advertising address used for the advertising in step 2.

17. Repeat steps 2–11 and 13–14, except that in step 10 the Lower Tester sends a LL_UNKNOWN_RSP PDU to the IUT with UnknownType set to LL_PERIODIC_SYNC_IND PDU and verifies that the IUT continues maintaining the connection.

18. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.

4.3.6.3 [Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]

- Test Purpose

Tests that the IUT with periodic advertising enabled is able to initiate the Periodic Advertising Sync Transfer Procedure. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well. The test purpose is exercised over each supported PHY, including using different PHYs for the connection and periodic advertising. The case that the remote does not support the procedure is tested.
The Lower Tester verifies that the IUT continues periodic advertising and that the IUT transfers synchronization information that is valid for that periodic advertising. The Upper Tester commands the IUT to perform the procedure and observes the responses returned by the IUT.

- **Reference**
  
  [13] 4.6.23

- **Initial Condition**
  
  MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

- **Test Procedure and Test Case IDs**
  
  Execute the test procedure using the parameters indicated in Table 4.64 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
<th>Primary ADV PHY</th>
<th>Secondary ADV PHY</th>
<th>Test Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-93-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>Yes</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-89-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-94-C</td>
<td>LE 1M</td>
<td>0x03 (LE Coded)</td>
<td>0x03 (LE Coded)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-90-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-95-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x02 (LE 2M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-91-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-96-C</td>
<td>LE Coded</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-92-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-97-C</td>
<td>LE 2M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-93-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.64: Values for each Test Case ID*
1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.64, and the Lower Tester connects with the IUT on that PHY.

2. The Lower Tester begins extended advertising using the PHY indicated in the Primary ADV PHY column in Table 4.64 for the primary advertising PHY, and the PHY indicated in the Secondary ADV PHY column in Table 4.64 for the secondary advertising PHY, using a public address.

3. The Lower Tester begins periodic advertising for the extended advertising in step 2.

4. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT with the Scanning_PHYs parameter set to the PHY indicated in the Primary ADV PHY column in Table 4.64 and Scan_Type[0] set to 0x00 (Passive Scanning).

5. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning.

Figure 4.295: [Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]
6. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval.
7. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester’s periodic advertisements.
8. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
9. The Upper Tester receives HCI_LE_Periodic_Advertising_Report events from the IUT.
10. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer command to the IUT with the Sync_Handle of the Lower Tester’s periodic advertising, Service_Data set to any non-zero value, and the Connection_Handle of the connection and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.
11. The Lower Tester receives an LL.PERIODIC_SYNC_IND PDU from the IUT with ID set to the value used for Service_Data in step 10, SyncInfo and connEventCount set to valid values, SID set to match the advertising set ID representing the advertising in step 2, AType set to the address type used for the advertising in step 2, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in Table 4.64, and the AdvA set to the Lower Tester’s advertiser address used for the advertising in step 2.
12. The Lower Tester verifies that the IUT continues maintaining the connection.
13. The Lower Tester uses the SyncInfo and connEventCount values to verify that they allow synchronization to the periodic advertising from step 3.
14. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate synchronization to periodic advertising.
15. The Upper Tester sends an HCI_LE_Set_Extended_Scanning_Enable command to the IUT to disable scanning.
16. The Lower Tester stops both extended advertising and periodic advertising.
17. If the Test Privacy column in Table 4.64 indicates No, then skip to step 19.
18. If the IUT supports controller-based privacy, then repeat steps 2–16, except that in step 2 the Lower Tester uses a resolvable private address matching an IRK in the IUT’s resolving list for the advertiser address, and that in step 11, the AdvA is set to the Lower Tester’s unresolved advertising address used for the advertising in step 2.
19. Terminate the connection between the IUT and the Lower Tester.

• Expected Outcome

Pass Verdict
- MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.
- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 6, the Advertising_SID value in the event is the value sent by the Lower Tester in step 2.

4.3.6.4 [Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYs]

• Test Purpose
Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure for each combination of periodic advertising PHYs and connection PHYs.

For various PHYs, the Lower Tester transmits periodic advertising and transfers synchronization information for that periodic advertising, and the Upper Tester verifies that the IUT synchronizes successfully to that periodic advertising and reports that periodic advertising. Additionally, it is verified that the IUT ignores the Periodic Advertiser List.
• Reference

• Initial Condition
  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.

• Test Procedure and Test Case IDs
  Execute the test procedure using the parameters indicated in Table 4.65 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
<th>Primary ADV PHY</th>
<th>Secondary ADV PHY</th>
<th>Test Periodic Advertiser List</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Step 1</td>
<td>Step 2</td>
<td>Step 2, Step 4</td>
<td>Step 8</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-98-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>Yes</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-94-C</td>
<td>LE 1M</td>
<td>0x03 (LE Coded)</td>
<td>0x03 (LE Coded)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-99-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x02 (LE 2M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-95-C</td>
<td>LE Coded</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-100-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-96-C</td>
<td>LE 2M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-97-C</td>
<td>LE Coded</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-102-C</td>
<td>LE 2M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-98-C</td>
<td>LE 1M</td>
<td>0x01 (LE 1M)</td>
<td>0x01 (LE 1M)</td>
<td>No</td>
</tr>
</tbody>
</table>

*Table 4.65: Values for each Test Case ID*
Figure 4.296: [Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYs]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.65, and the Lower Tester connects with the IUT on that PHY.

2. The Lower Tester begins periodic advertising using the PHY indicated in the Secondary ADV PHY column in Table 4.65 and any periodic interval. The Lower Tester may begin extended advertising using the PHY indicated in the Primary ADV PHY column in Table 4.65 for the primary advertising PHY, and the PHY indicated in the Secondary ADV PHY column in Table 4.65 for the secondary advertising PHY, using a public address; if so, this refers to the periodic advertising.

3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

4. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount and SyncInfo set to any values that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in Table 4.65, and the AdvA set to the Lower Tester’s advertiser address used for the advertising in step 2.
5. The Upper Tester receives an HCI_LE_Periodic_Advertising.Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester, and the Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.

6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate.Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

7. The Lower Tester stops periodic advertising and extended advertising (if started).

8. If the Test Periodic Advertiser List column in Table 4.65 indicates No, then skip to step 12.

9. The Upper Tester sends an HCI_LE_Add_Device_To_Periodic_Advertiser_List command to the IUT, containing an advertiser address different than the one used by the Lower Tester for periodic advertising, and receives an HCI_Command_Complete event in response.

10. Repeat steps 2–7, expecting the IUT to ignore the Periodic Advertiser List.

11. The Upper Tester sends an HCI_LE_Clear_Periodic_Advertiser_List command to the IUT and receives an HCI_Command_Complete event in response.

12. Terminate the connection between the IUT and the Lower Tester.

• Expected Outcome

  Pass Verdict

  MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising, and generates reports after synchronization.

4.3.6.5 [Periodic Advertising Sync Transfer Procedure, Accepting – Skipping Events]

• Test Case IDs

  LL/CON/SLA/BV-103-C
  LL/CON/MAS/BV-99-C

• Test Purpose

  Tests that the IUT is able to send reports for a periodic advertising after synchronization at least as often as allowed by the Skip value.

• Reference


• Initial Condition

  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.
1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.

2. Execute steps 3-9 for each round shown in Table 4.66.

<table>
<thead>
<tr>
<th>Round</th>
<th>Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0x0000</td>
</tr>
<tr>
<td>2</td>
<td>0x0001</td>
</tr>
<tr>
<td>3</td>
<td>0x0013</td>
</tr>
</tbody>
</table>

Table 4.66: Values for each case variation.

3. The Lower Tester begins periodic advertising using the LE 1M PHY and a periodic interval greater than: (the max offset describable in a SyncInfo) + 2 x (the connection interval). The periodic advertising data for each periodic advertising event contains the event number, to allow the Upper Tester to determine exactly which periodic advertising events the IUT is reporting. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.

4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to a value as specified in Table 4.66 for this round, and Sync_Timeout set to (Skip + 3) x the
periodic advertising interval and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

5. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount set to the nearest connection event for which a valid offset in the SyncInfo can be specified to point to the periodic advertising, SyncInfo set to the value that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the Lower Tester’s advertiser address used for the advertising in step 3.

6. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester.

7. The Upper Tester receives at least one HCI_LE_Periodic_Advertising_Report event from the IUT for every (1 + Skip) periodic events; repeat until at least 30 sets of (1+Skip) periodic events have passed.

8. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

9. The Lower Tester stops periodic advertising and extended advertising (if started).

10. Terminate the connection between the IUT and the Lower Tester.

• Expected Outcome

  Pass Verdict
  MAS/SLA: In each case, the IUT generates reports after synchronization at least as often as one report for every (1 + Skip) periodic events. The event numbers reported to the Upper Tester are no more than (1 + Skip) apart; two omissions out of the 30 repetitions are allowed, in order to account for packet loss.

4.3.6.6 [Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]

• Test Case IDs

  LL/CON/SLA/BV-104-C
  LL/CON/MAS/BV-100-C

• Test Purpose

  Tests that the IUT synchronized to periodic advertising does not report synchronizing a second time if it receives synchronization information for the same periodic advertising.

• Reference


• Initial Condition

  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.
### Test Procedure

1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
2. The Lower Tester begins periodic advertising using the LE 1M PHY and any periodic interval. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.
3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.
4. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount and SyncInfo set to any values that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the Lower Tester’s advertiser address used for the advertising in step 2.
5. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU.
6. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.

---

**Figure 4.298: [Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]**

1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
2. The Lower Tester begins periodic advertising using the LE 1M PHY and any periodic interval. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.
3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.
4. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount and SyncInfo set to any values that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the Lower Tester’s advertiser address used for the advertising in step 2.
5. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU.
6. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
7. Repeat step 4 two more times, with parameters describing the same periodic advertising. On each execution of this step, the Upper Tester does not receive any additional HCI_LE_Periodic_Advertising_Sync_Transfer_Received event.

8. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester receives at least 10 more HCI_LE_Periodic_Advertising_Report events from the IUT.

9. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate.Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

10. The Lower Tester stops periodic advertising and extended advertising (if started).

11. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

**Pass Verdict**

MAS/SLA: The IUT does not generate a second HCI_LE_Periodic_Advertising_Sync_Transfer_Received event for the same periodic advertising, but continues to generate HCI_LE_Periodic_Advertising_Report events with the same Sync_Handle.

4.3.6.7 [Periodic Advertising Sync Transfer Procedure, Accepting – Extreme Timings]

- Test Case IDs

  LL/CON/SLA/BV-105-C
  LL/CON/MAS/BV-101-C

- Test Purpose

  Tests that the IUT is able to synchronize to periodic advertising using connEventCount values that represent short and large distances into the past and future.

- Reference


- Initial Condition

  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.
**Test Procedure**

1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
2. Execute steps 3-9 for each round shown in Table 4.67.

**Note:** connEventCount does not need to be an event that has occurred.

<table>
<thead>
<tr>
<th>Round</th>
<th>Lower Tester's Periodic Advertising Interval</th>
<th>connEventCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td>greater than (the max offset describable in a SyncInfo) + 2 x connection interval</td>
<td>Step 5</td>
</tr>
<tr>
<td>1</td>
<td>greater than (the max offset describable in a SyncInfo) + 2 x connection interval</td>
<td>Minimum distance in the future</td>
</tr>
<tr>
<td>2</td>
<td>greater than (the max offset describable in a SyncInfo) + 2 x connection interval</td>
<td>Maximum distance in the future</td>
</tr>
<tr>
<td>3</td>
<td>greater than (the max offset describable in a SyncInfo) + 2 x connection interval</td>
<td>Minimum distance in the past</td>
</tr>
<tr>
<td>4</td>
<td>greater than (the max offset describable in a SyncInfo) + 2 x connection interval</td>
<td>Maximum distance in the past</td>
</tr>
<tr>
<td>5</td>
<td>less than half the connection interval</td>
<td>Maximum distance in the past</td>
</tr>
<tr>
<td>6</td>
<td>less than half the connection interval</td>
<td>Minimum distance in the future</td>
</tr>
</tbody>
</table>

*Table 4.67: Values for each case variation.*
3. The Lower Tester begins periodic advertising using the LE 1M PHY and a periodic interval as specified in Table 4.67. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.

4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

5. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values as specified in Table 4.67 for this round that would allow synchronization to its periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address. In Table 4.67, “Minimum distance” means the nearest connection event, and “Maximum distance” means the furthest connection event, in the specified direction for which a valid offset in the SyncInfo can be specified to point to the periodic advertising.

6. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU.

7. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.

8. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

9. The Lower Tester stops periodic advertising and extended advertising (if started).

10. Terminate the connection between the IUT and the Lower Tester.

- **Expected Outcome**

  **Pass Verdict**

  MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising and generates reports after synchronization.

4.3.6.8  **[Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure]**

- **Test Case IDs**
  
  LL/CON/SLA/BV-106-C
  LL/CON/MAS/BV-102-C

- **Test Purpose**

  Tests that the IUT is able to report synchronization failure in a timely manner.

- **Reference**


- **Initial Condition**

  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with an IRK in the resolving list for resolving a resolvable private address used in advertising from the Lower Tester.
• **Test Procedure**

![Diagram of the test procedure](image)

*Figure 4.300: [Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure]*

1. **The Upper Tester** configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY. The periodic advertising interval should be around three times the connection interval.

2. **The Upper Tester** sends an `HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters` command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to at least 12 periodic advertising intervals, and receives an `HCI_Command_Complete` with Status set to 0x00 (Success) in response.

3. **The Lower Tester** sends an `LL_PERIODIC_SYNC_IND` PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address.

4. After 6 periodic advertising intervals have passed, starting with the moment where the next periodic event would be expected, the Upper Tester receives an `HCI_LE_Periodic_Advertising_Sync_Transfer_Received` event from the IUT with Status set to Connection Failed to be Established / Synchronization Timeout (0x3E).

5. From the 7th periodic advertising event onwards, the Lower Tester sends `AUX_SYNC_IND` PDUs to the IUT at times corresponding to the data in the LL_PERIODIC_SYNC_IND PDUs.

• **Expected Outcome**

**Pass Verdict**

MAS/SLA: The IUT reports a failure to synchronize after it does not receive any periodic advertising packet after 6 periodic advertising intervals.

**Fail Verdict**

MAS/SLA: The IUT sends an `HCI_LE_Periodic_Advertising_Report` event to the Upper Tester.
4.3.6.9 [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses]

- Test Case IDs
  LL/CON/SLA/BV-107-C
  LL/CON/MAS/BV-103-C

- Test Purpose
  Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure for each Periodic Advertising Sync Transfer Mode. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well.

  The Lower Tester transmits periodic advertising and transfers synchronization information for that periodic advertising, and the Upper Tester verifies that the IUT synchronizes (when enabled) successfully to that periodic advertising and reports (when enabled) that periodic advertising. For transfer modes including synchronization, enabling and disabling of reporting is tested. It is tested that the IUT uses the configured transfer mode for the connection instead of the default transfer mode is tested.

- Reference

- Initial Condition
  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with an IRK in the resolving list for resolving a resolvable private address used in advertising from the Lower Tester.

  The IUT shall not have performed the
  HCI_LE_Set_Default_Periodic_Advertising_Sync_Transfer_Parameters command since last being reset.
Figure 4.301: [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses] part 1 of 2
The Upper Tester expects at least 10 events, then skip to step 17.

**STEP 14**

The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report_Event for at least 10 periodic advertising intervals.

HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters

(Mode = 0x02)

HCI_Command_Complete_Event

(Status: 0x00)

The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report_Event for at least 10 periodic advertising intervals.

Repeat two times

**STEP 17**

HCI_LE_Set_Periodic_Advertising_Receive_Enable

(Enable = 0x00)

HCI_Command_Complete_Event

(Status: 0x00)

The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report_Event for at least 10 periodic advertising intervals.

The Upper Tester expects at least 10 events.

**STEP 23**

If the IUT supports controller-based privacy, then repeat steps 3–11 using resolvable private addresses for the Lower Tester.

Figure 4.302: [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses] part 2 of 2

1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
2. Execute steps 3–23 for each round shown in Table 4.68.

<table>
<thead>
<tr>
<th>Round</th>
<th>Default Mode</th>
<th>New connection</th>
<th>Specific Mode</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>Step 5</td>
<td>Step 6</td>
<td>Steps 8 &amp; 10</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>0x00</td>
</tr>
</tbody>
</table>
3. The Lower Tester begins periodic advertising using the LE 1M PHY and any periodic interval. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising. The Lower Tester uses a different advertising set ID for each round.

4. If a value is specified for Default Mode in Table 4.68 for this round, the Upper Tester sends an HCI_LE_Set_Default_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to the value specified for Default Mode in Table 4.68 for this round, Skip set to 0, and Sync_Timeout set to 30 times the periodic advertising interval set in step 3, and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

5. If "Yes" is specified for New Connection in Table 4.68 for this round, terminate the connection between the IUT and the Lower Tester and then re-create it as in step 1.

6. If "Yes" is specified for Specific Mode in Table 4.68 for this round, the Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to the value specified for Mode in Table 4.68 for this round, Skip set to 0, and Sync_Timeout set to 30 times the periodic advertising interval, and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

7. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values that would allow synchronization to its periodic advertising, SID set to the value used in step 4, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the address used for the extended advertising in step 3 (or any valid public address if the Lower Tester is not advertising).

8. If the Mode specified in Table 4.68 for this round is 0x00 (no synchronization), the Upper Tester receives no HCI_LE_Periodic_Advertising_Sync_Transfer_Received or HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising interval, then skip to step 23.

9. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester.

10. If the Mode specified in Table 4.68 for this round is 0x01 (synchronize with reporting initially disabled), skip to step 14.

11. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.

12. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x01, and all

<table>
<thead>
<tr>
<th>Round</th>
<th>Default Mode</th>
<th>New connection</th>
<th>Specific Mode</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>0x02</td>
<td>No</td>
<td>No</td>
<td>0x00</td>
</tr>
<tr>
<td>Step 5</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>0x02</td>
</tr>
<tr>
<td>Step 6</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
<td>0x00</td>
</tr>
<tr>
<td>Steps 8 &amp; 10</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
<td>0x01</td>
</tr>
</tbody>
</table>

Table 4.68: Values for each case variation.
other parameters set to the values specified in step 6, and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

13. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT, then skip to step 17.

14. The Upper Tester receives no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

15. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, and all other parameters set to the values specified in step 6, and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

16. The Upper Tester receives no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

17. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x00 and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

18. The Upper Tester receives no further HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

19. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x01 and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

20. The Upper Tester receives at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.


22. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and receives an HCI_Command_Complete event in response.

23. The Lower Tester stops periodic advertising and extended advertising (if started).

24. If the IUT supports controller-based privacy, then repeat steps 3–11 using the values for round 6 in Table 4.68, except that in step 3 the Lower Tester uses a resolvable private address corresponding to a public identity address and matching an IRK in the IUT’s resolving list for the advertiser address in the extended advertising, if started, that in step 7 the AdvA is set to the Lower Tester’s unresolved advertising address, and that in step 9 the HCI_LE_Periodic_Advertising_Sync_Transfer_Received event the Advertiser_Address_Type is set to 0x02 (Public Identity Address (corresponds to Resolved Private Address)) and the Advertiser_Address is set to the Lower Tester’s resolved advertiser address.

25. If the IUT supports controller-based privacy, then repeat steps 3–11 using the values for round 6 in Table 4.68, except that in step 3 the Lower Tester uses a resolvable private address not matching an IRK in the IUT’s resolving list for the advertiser address in the extended advertising, if started, that in step 7 the AdvA is set to the Lower Tester’s unresolved advertising address, and that in step 9 the HCI_LE_Periodic_Advertising_Sync_Transfer_Received event the Advertiser_Address_Type is set to 0x01 (Random Device Address / Random (static) Identity Address) and the Advertiser_Address is set to the Lower Tester’s unresolved advertiser address.

26. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising only when configured by the transfer mode, and generates reports after synchronization only when configured by the transfer mode or when reporting is enabled.
4.3.6.10 [Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]

- **Test Case IDs**
  - LL/CON/SLA/BV-108-C
  - LL/CON/MAS/BV-104-C

- **Test Purpose**
  Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure and synchronize successfully when the Transfer Mode is changed while synchronization is in progress.

- **Reference**

- **Initial Condition**
  MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.

- **Test Procedure**

  ![Diagram of the test procedure]

  **Figure 4.303:** [Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]

  1. The Lower Tester begins periodic advertising using the LE 1M PHY with a periodic advertising interval large enough to support the timing requirements in steps 4–6. The Lower Tester may
begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.

2. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to 0x01 (synchronize with reporting initially disabled), Skip set to 0, and Sync_Timeout set to any value and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

4. At a time when the next periodic advertising event would occur far enough in the future to allow completing this step and step 6 before that periodic advertising event is started, the Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values that would allow synchronization to its periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address. The Lower Tester receives an LL data PDU from the IUT, acknowledging the receipt of the LL_PERIODIC_SYNC_IND PDU.

5. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to 0x02 (synchronize with reporting initially enabled), Skip set to 0, and Sync_Timeout set to any value and receives an HCI_Command_Complete with Status set to 0x00 (Success) in response.

6. The Upper Tester receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the advertisements generated by the Lower Tester.

7. The Upper Tester receives no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.

- Expected Outcome

Pass Verdict

MAS/SLA: The IUT accepts periodic advertising sync information and synchronizes to periodic advertising following the transfer mode used during reception of the periodic advertising sync information.

4.3.6.11 [Acknowledging Long Control PDUs]

- Test Purpose

Tests that the IUT is able to receive a long LL Control PDU, acknowledge it, and continue maintaining the connection. The test purpose is exercised over each supported PHY.

- Reference

[13] 4.5.11

- Initial Condition

MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in Table 4.69 for each applicable Test Case ID.
<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-109-C</td>
<td>LE 1M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-105-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-110-C</td>
<td>LE 2M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-106-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-111-C</td>
<td>LE Coded</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-107-C</td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.69: Values for each Test Case ID*

![Flowchart](image)

*Figure 4.304: [Acknowledging Long Control PDUs]*

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in *Table 4.69*, and the Lower Tester connects with the IUT on that PHY.
2. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum value (27 octets).
3. The Lower Tester transmits any LL Control PDU of a length supported by the IUT for reception that is greater than 27 octets.
4. The Lower Tester expects the IUT to send an LL acknowledgement and to continue maintaining the connection for at least 10 connection events.
5. Terminate the connection between the IUT and the Lower Tester.

- **Expected Outcome**

  **Pass Verdict**

  MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.

- **Notes**

  To determine the maximum LL Control PDU length supported by the IUT (for a received PDU), a value that is required in step 3 of the test procedure, the tester may perform an LL feature exchange procedure or it may use the information available in the ICS.
4.3.6.12  [Rejecting Request to Send Long Control PDUs before Feature Exchange]

- **Test Purpose**
  Tests that the IUT does not send a long LL Control PDU before feature exchange, including when limits set by Data PDU Length Management allow sending LL Data PDUs of equal or greater length (if supported). The test purpose is exercised over each supported PHY.

- **Reference**
  [13] 4.5.11

- **Initial Condition**
  MAS/SLA: The IUT is not maintaining any connections.

- **Test Procedure and Test Case IDs**
  Execute the test procedure using the parameters indicated in Table 4.70 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-112-C</td>
<td>LE 1M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-108-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-113-C</td>
<td>LE 2M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-109-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-114-C</td>
<td>LE Coded</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-110-C</td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.70: Values for each Test Case ID*

*Figure 4.305: [Rejecting Request to Send Long Control PDUs before Feature Exchange]*
1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.70, and the Lower Tester connects with the IUT on that PHY.

2. If the IUT autonomously initiates feature exchange at any time between step 1 and step 9, the test ends with an Inconclusive Verdict.

3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum values (27 octets and 328 us).

4. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU (containing a CtrData field of 27 octets or larger).

5. The Upper Tester expects the IUT to reject the command.

6. If the IUT does not support the Data Length Update Procedure, skip steps 7–9.

7. The Lower Tester initiates the Data Length Update Procedure with the maximum values and expects the IUT to complete the procedure.

8. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU of length equal to or less than the maximum length of LL Data PDU that the Lower Tester indicated support for reception in the previous step.

9. The Upper Tester expects the IUT to reject the command.

10. Terminate the connection between the IUT and the Lower Tester.

**Expected Outcome**

**Pass Verdict**

MAS/SLA: The IUT rejects commands to initiate procedures involving long LL Control PDUs before feature exchange.

4.3.6.13 [Rejecting Request to Send Long Control PDUs after Feature Exchange]

**Test Purpose**

Tests that the IUT does not send a long LL Control PDU after feature exchange indicating lack of support of any features using LL Control PDUs of that length or greater by the Lower Tester, including when limits set by Data PDU Length Management allow sending LL Data PDUs of equal or greater length (if supported). The test purpose is exercised over each supported PHY.

**Reference**

[13] 4.5.11

**Initial Condition**

MAS/SLA: The IUT is not maintaining any connections.

**Test Procedure and Test Case IDs**

Execute the test procedure using the parameters indicated in Table 4.71 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Step 1</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-115-C</td>
<td>LE 1M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-111-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-116-C</td>
<td>LE 2M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-112-C</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4.71: Values for each Test Case ID

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-117-C</td>
<td>LE Coded</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-113-C</td>
<td></td>
</tr>
</tbody>
</table>

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.71, and the Lower Tester connects with the IUT on that PHY.
2. The Lower Tester initiates feature exchange or responds to feature exchange, if autonomously initiated by the IUT, indicating no support for any procedures involving long LL Control PDUs.
3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum values (27 octets and 328 us).
4. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU (containing a CtrData field of 27 octets or larger).
5. The Upper Tester expects the IUT to reject the command.
6. If the IUT does not support the Data Length Update Procedure, skip steps 7–9.
7. The Lower Tester initiates the Data Length Update Procedure with the maximum values and expects the IUT to complete the procedure.
8. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU of length equal to or less than the maximum length of LL Data PDU that the Lower Tester indicated support for reception in the previous step.
9. The Upper Tester expects the IUT to reject the command with the error code 0x1A (Unsupported Remote Feature).
10. Terminate the connection between the IUT and the Lower Tester.
• Expected Outcome

Pass Verdict

MAS/SLA: The IUT rejects commands to initiate procedures involving long LL Control PDUs after feature exchange indicating that the Lower Tester does not support any such procedures.

4.3.6.14 [Sending Long Control PDUs after Feature Exchange]

• Test Purpose

Tests that the IUT is able to send a long LL Control PDU after feature exchange indicating support of any features using LL Control PDUs of that length or greater by the Lower Tester, including when limits set by Data PDU Length Management disallow sending LL Data PDUs of that length, and continue maintaining the connection. The test purpose is exercised over each supported PHY.

• Reference

[13] 4.5.11

• Initial Condition

MAS/SLA: The IUT is not maintaining any connections.

• Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in Table 4.72 for each applicable Test Case ID.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Connection PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-118-C</td>
<td>LE 1M</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-114-C</td>
<td>LE 2M</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-119-C</td>
<td>LE Coded</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-115-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/SLA/BV-120-C</td>
<td></td>
</tr>
<tr>
<td>LL/CON/MAS/BV-116-C</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.72: Values for each Test Case ID
1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in Table 4.72, and the Lower Tester connects with the IUT on that PHY.

2. The Lower Tester initiates feature exchange or responds to feature exchange, if autonomously initiated by the IUT, indicating support for at least one procedure involving long LL Control PDUs that the IUT supports.

3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum value (27 octets).

4. The Upper Tester commands the IUT to initiate any procedure for which the Lower Tester indicated support that involves the IUT transmitting a long LL Control PDU (i.e., payload length larger than 27 octets).

5. The Lower Tester expects the IUT to initiate the procedure.

6. The Lower Tester continues the procedure at least until the IUT has sent the expected, valid long PDU.

7. Terminate the connection between the IUT and the Lower Tester.

- **Expected Outcome**
  - **Pass Verdict**

    MAS/SLA: The IUT sends long LL Control PDUs after feature exchange indicating that the Lower Tester supports them.

**4.3.6.15 [Constant Tone Extension Request Procedure, IUT Initiated, Unsupported]**

- **Test Purpose**

  Tests that the IUT correctly handles the case where the remote does not support the Connection CTE Response feature.

  The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and confirms the IUT correctly handles the LL_UNKNOWN_RSP PDU generated by the Lower Tester. When aware of the missing feature support on the remote Lower Tester, the IUT will not initiate a CTE Request Procedure.
• Test Case IDs
  LL/CON/SLA/BV-63-C
  LL/CON/MAS/BV-59-C

• Reference
  [10] 5.1.12

• Initial Condition
  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

• Test Procedure
  Execute the test procedure for IUT as Master (if supported) and IUT as Slave (if supported). If the Lower Tester has exchanged features with the IUT, skip steps 2–5.

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).

2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).

3. The Lower Tester receives an LL_CTE_REQ PDU.

4. The Lower Tester sends an LL_UNKNOWN_RSP PDU. UnknownType is set to LL_CTE_REQ.

Figure 4.308: [Constant Tone Extension Request Procedure, IUT Initiated, Unsupported]
5. The Upper Tester receives an HCI_LE_CTE_Request_Failed event from the IUT. Status shall be set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).

6. Repeat step 2.

- Expected Outcome

  **Pass Verdict**

  On step 2, the IUT sends an LL_CTE_REQ PDU. The Upper Tester receives an HCI_LE_CTE_Request_Failed event with the Status set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).

  On step 6, The IUT does not send any LL_CTE_REQ PDU. The Upper Tester receives an HCI_Command_Complete event with the Status set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).

4.3.6.16  **Unrequested Constant Tone Extension, IUT Receiving, AoD**

- **Test Purpose**

  Tests that an IUT can process an unrequested AoD Constant Tone Extension from a remote Lower Tester. Test that the IUT sends the appropriate event to the Upper Tester based on the Lower Tester's Constant Tone Extension.

  The Lower Tester maintains a connection and transmits an AoD Constant Tone Extension on a data packet. The Upper Tester observes the event returned by the IUT.

- **Reference**

  [13] 2.5.4, 5.1.12

- **Initial Condition**

  Connected role as specified in Table 4.74 (PHY as specified in Table 4.74, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value)
• Test Procedure

For each round from 1 to 3 based on Table 4.73:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime (Step 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>0x14</td>
</tr>
</tbody>
</table>

Table 4.73: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x01 (enable).
2. The Lower Tester sends a data channel PDU to the IUT. The data channel PDU contains the CTETInfo field, with CTETime set to the value specified in Table 4.73, RFU set to ‘0’, and the CTEType set to 2 (AoD Constant Tone Extension with 2 µs slots). The packet containing the data channel PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the specified CTETime.

3. The Upper Tester receives an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the data channel PDU was received, and RSSI shall be set to a valid value. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots). The connEventCounter shall be set to the connection event counter value of the data packet PDU.

4. Repeat steps 2–3 for each round shown in Table 4.73.

5. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x00 (disable).

6. Repeat step 2.

7. The Upper Tester does not receive an HCI_LE_Connection_IQ_Report event from the IUT.

8. The Lower Tester expects the IUT to maintain the connection.

9. If the IUT supports 1 µs slots, repeat steps 1–4, except that in step 2 CTEType is set to 1 (AoD Constant Tone Extension with 1 µs slots) and that in step 3 CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 µs slots).

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6.16.2 LL/CON/SLA/BV-70-C [Unrequested Constant Tone Extension, IUT Receiving, AoD – LE 1M PHY]</td>
<td>Slave</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.16.3 LL/CON/MAS/BV-118-C [Unrequested Constant Tone Extension, IUT Receiving, AoD – LE 2M PHY]</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.3.6.16.4 LL/CON/SLA/BV-121-C [Unrequested Constant Tone Extension, IUT Receiving, AoD – LE 2M PHY]</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
</tbody>
</table>

Table 4.74: Unrequested Constant Tone Extension, IUT Receiving, AoD Test Cases

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT generates an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x01.
- The IUT does not generate an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x00.
- The IUT acknowledges the Data Channel PDUs containing a Constant Tone Extension sent by the Lower Tester.
4.3.6.17 Constant Tone Extension Request Procedure, IUT Initiated, AoA

- **Test Purpose**
  Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

  The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- **Reference**
  [10] 2.5, 5.1.12

- **Initial Condition**
  Connected role as specified in Table 4.76 (PHY as specified in Table 4.76, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

  The antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- **Test Procedure**

  ![Diagram of the test procedure](image)

  **Figure 4.310: Constant Tone Extension Request Procedure, IUT Initiated, AoA**

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_Length_Switching_Pattern between 0x02 and 0x4B. The Upper Tester stores the Max_Length_Switching_Pattern for future use.
For each round from 1–3 based on Table 4.75:

<table>
<thead>
<tr>
<th>Round</th>
<th>Requested_CTE_Length (Step 3, 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>0x14</td>
</tr>
</tbody>
</table>

*Table 4.75: Parameter values for each case variation*

2. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).

3. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to the value specified in Table 4.75. Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension).

4. The Lower Tester receives an LL_CTE_REQ PDU. MinCTELenReq shall be set to the Requested_CTE_Length value from step 3. CTETypeReq shall be set to 0 (AoA Constant Tone Extension).

5. The Lower Tester sends an LL_CTE_RSP PDU. The LL_CTE_RSP PDU contains the CTEInfo field, with CTETime set to the MinCTELenReq value from step 4, RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in Table 4.75.

6. The Upper Tester receives an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the LL_CTE_RSP PDU was received, and RSSI shall be set to a valid value. RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array from step 2. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). Slot_Durations shall be set to 0x02 (2 µs slots). The connEventCounter shall be set to the connection event counter value of the LL_CTE_RSP PDU. Sample_Count shall be set to 8 + (8 x CTETime - 12) / 4. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.

7. Repeat steps 2–6 for each round shown in Table 4.75.

8. If the IUT supports 1 µs slots, repeat steps 2–7, except that in step 2 Slot_Durations shall be set to 0x01 (1 µs slots) and that in step 6 Sample_Count shall be set to 8 + (8 x CTETime - 12) / 2.
• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6.17.1 LL/CON/MAS/BV-56-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 1M PHY]</td>
<td>Master</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.17.2 LL/CON/SLA/BV-60-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 1M PHY]</td>
<td>Slave</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.17.3 LL/CON/MAS/BV-119-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 2M PHY]</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.3.6.17.4 LL/CON/SLA/BV-122-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 2M PHY]</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
</tbody>
</table>

Table 4.76: Constant Tone Extension Request Procedure, IUT Initiated, AoA Test Cases

• Expected Outcome

  Pass Verdict
  For all rounds described in the test procedure, the following condition shall occur:
  - The IUT generates an HCI_LE_Connection_IQ_Report event.

  4.3.6.18 [Constant Tone Extension Request Procedure, IUT Initiated, Periodic]

  • Test Purpose

  Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when periodic requests are selected. Test that the IUT generates multiple CTE Requests at the correct interval and returns appropriate responses to the Upper Tester based on the Lower Tester responses.

  The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates periodic CTE Requests, and observes the responses returned by the IUT.

• Reference

  [10] 2.5, 5.1.12

• Test Case IDs

  LL/CON/SLA/BV-61-C
  LL/CON/MAS/BV-57-C

• Initial Condition

  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
### Test Procedure

**Figure 4.311: [Constant Tone Extension Request Procedure, IUT Initiated, Periodic]**

For each round from 1 to 3 based on **Table 4.77**:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTE_Request_Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x0001</td>
</tr>
<tr>
<td>2</td>
<td>0x0010</td>
</tr>
<tr>
<td>3</td>
<td>0x0167</td>
</tr>
<tr>
<td>4</td>
<td>A randomly selected value between 0x0080 (ca 5 s) and 0x0200 (ca 20 s) inclusive</td>
</tr>
</tbody>
</table>

**Table 4.77: Parameter values for each case variation**

1. The Upper Tester sends an **HCI_LE_Set_Connection_CTE_Receive_Parameters** command to the IUT. **Connection_Handle** shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception **Length_of_Switching_Pattern** shall be set to 0x02, **Antenna_IDs[0]** and **Antenna_IDs[1]** shall be set to the pattern 0, 1, and **Slot_Durations** shall be set to 0x02 (2 µs slots). **Sampling_Enable** shall be set to 0x01 (enable).
2. The Upper Tester sends an **HCI_LE_Connection_CTE_Request_Enable** command to the IUT. **Connection_Handle** shall be set to the handle of the connection. **Enable** shall be set 0x01 (Enable). **CTE_Request_Interval** shall be set to the value specified in **Table 4.77**. **Requested_CTE_Length** shall be set to 0x02. If the IUT supports AoA Constant Tone Extension
reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).

3. The Lower Tester receives an LL_CTE_REQ PDU.

4. The Lower Tester sends an LL_CTE_RSP PDU. The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in step 2.

5. The Upper Tester receives an HCI_LE_Connection_IQ_Report event from the IUT.

6. Repeat steps 3–5 until 5 Constant Tone Extensions have been transmitted. The Lower Tester receives each successive LL_CTE_RSP PDU after a number of connection events equal to or greater than the CTE_Request_Interval value in Table 4.77 since receiving the last such PDU.

7. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x00 (Disable).

8. Repeat steps 1–7 for each round shown in Table 4.77.

9. If the IUT supports 1 µs slots, repeat steps 1–8, except that in step 1 Slot_Durations shall be set to 0x01 (1 µs slots) and that in step 5 Slot_Durations shall be set to 0x01 (1 µs slots).

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT periodically requests Constant Tone Extensions, with at least CTE_Request_Interval connection events between requests.

- The IUT generates an HCI_LE_Connection_IQ_Report event.

4.3.6.19 [Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]

• Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT correctly handles the case where the remote Lower Tester has Connection CTE Responses disabled.

The Lower Tester maintains a connection and has disabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and confirms the IUT correctly handles an LL_REJECT_EXT_IND PDU.

• Reference

[10] 5.1.12

• Test Case IDs

LL/CON/SLA/BV-62-C
LL/CON/MAS/BV-58-C

• Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
• Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).

2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).

3. The Lower Tester receives an LL_CTE_REQ PDU.

4. The Lower Tester sends an LL_REJECT_EXT_IND PDU. RejectOpCode is set to LL_CTE_REQ. ErrorCode is set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

5. The Upper Tester receives an HCI_LE_CTE_Request_Failed event from the IUT. Status shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

• Expected Outcome

Pass Verdict
The IUT generates an HCI_LE_CTE_Request_Failed event with the Status set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

4.3.6.20 [Constant Tone Extension Request Procedure, IUT Initiated, Timeout]

• Test Purpose
Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT correctly handles the case where the remote Lower Tester does not respond and the procedure times out.
The Lower Tester maintains a connection. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request. The Lower Tester ignores the request and allows the procedure to time out.

- Reference

[10] 5.1.12

- Test Case IDs

LL/CON/SLA/BV-64-C  
LL/CON/MAS/BV-60-C

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

![Test Procedure Diagram]

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).

2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).

3. The Lower Tester receives an LL_CTE_REQ PDU.

4. The Lower Tester does not respond, allowing the procedure to time out.

5. The Upper Tester receives an HCI_Disconnect_Complete event from the IUT with reason code set to 0x22 (LL Response Timeout).

6. The Lower Tester expects the IUT to stop maintaining the connection.

Figure 4.313: [Constant Tone Extension Request Procedure, IUT Initiated, Timeout]
• **Expected Outcome**

  **Pass Verdict**
  The IUT disconnects the link after a procedure response timeout.

  **4.3.6.21 [Constant Tone Extension Request Procedure, IUT Responding, AoA]**

• **Test Purpose**

  Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

  The Lower Tester maintains a connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT’s response for proper formatting.

• **Reference**

  [10] 2.5, 5.1.12

• **Test Case IDs**

  LL/CON/SLA/BV-65-C  
  LL/CON/MAS/BV-61-C

• **Initial Condition**

  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
• Test Procedure

![Diagram of test procedure]

Figure 4.314: [Constant Tone Extension Request Procedure, IUT Responding, AoA]

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_CTE_Length for future use.

For each round from 1 to 3 based on Table 4.78:

<table>
<thead>
<tr>
<th>Round</th>
<th>Requested_CTE_Length (Step 2, 5-6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>Max_CTE_Length</td>
</tr>
</tbody>
</table>

Table 4.78: Parameter values for each case variation

2. If the CTETime listed in Table 4.78 for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 8.

3. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Transmit_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. CTE_Types shall be set to Allow AoA CTE Response (0000001b).

4. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).
5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to the Requested_CTE_Length value specified in Table 4.78. CTETypeReq is set to 0 (AoA Constant Tone Extension).
6. The Lower Tester receives an LL_CTE_RSP PDU. The LL_CTE_RSP PDU shall contain the CTEInfo field, with CTETime set to a value greater than or equal to the MinCTELenReq value from step 5, RFU set to ‘0’, and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing an LL_CTE_RSP PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime.
7. The Lower Tester sends an Empty PDU to the IUT not acknowledging the LL_CTE_RSP PDU.
8. The Lower Tester receives an additional LL_CTE_RSP PDU with the same fields as in step 6.
9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).
10. Repeat steps 1–9 for each round shown in Table 4.78.

• Expected Outcome
  
  Pass Verdict
  
  For all rounds described in the test procedure, the following condition shall occur:
  - The Lower Tester receives the CTE Response.

4.3.6.22  [Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]

• Test Purpose
  
  Tests that an IUT responds correctly to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses have been disabled. Test that the IUT generates an LL_REJECT_EXT_IND PDU.

  The Lower Tester maintains a connection. The Upper Tester configures the IUT to disable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and confirms the IUT correctly sends an LL_REJECT_EXT_IND PDU.

• Reference
  
  [10] 5.1.12

• Test Case IDs
  
  LL/CON/SLA/BV-66-C
  LL/CON/MAS/BV-62-C

• Initial Condition
  
  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

  The IUT does not have CTE Responses enabled for the connection.
• **Test Procedure**

![Diagram showing the test procedure](image)

- **Lower Tester** sends an LL_CTE_REQ PDU. MinCTELenReq is set to the 0x02. If the IUT supports AoA CTE Transmission, CTETypeReq is set to 0 (AoA Constant Tone Extension); otherwise CTETypeReq is set to 2 (AoD Constant Tone Extension with 2 us slots).

- **Lower Tester** receives an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

- **Expected Outcome**

  **Pass Verdict**

  The IUT sends an LL_REJECT_EXT_IND PDU with RejectOpCode set to LL_CTE_REQ and ErrorCode set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

4.3.6.23 **Constant Tone Extension Request Procedure, IUT Initiated, AoD**

- **Test Purpose**

  Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. The IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

  The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- **Reference**

  [10] 2.5, 5.1.12

- **Initial Condition**

  Connected role as specified in Table 4.80 (PHY as specified in Table 4.80, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
• Test Procedure

For each round from 1 to 3 based on Table 4.79:

<table>
<thead>
<tr>
<th>Round</th>
<th>Requested_CTE_Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>0x14</td>
</tr>
</tbody>
</table>

Table 4.79: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x01 (enable).
2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to the value specified in Table 4.79. Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).
3. The Lower Tester receives an LL_CTE_REQ PDU. MinCTELenReq shall be set to the Requested_CTE_Length value from step 2. CTETypeReq shall be set to 2 (AoD Constant Tone Extension with 2 µs slots).
4. The Lower Tester sends an LL_CTE_RSP PDU. The LL_CTE_RSP PDU contains the CTEInfo field, with CTETime set to the MinCTELenReq value from step 3, RFU set to ‘0’, and the CTEType set to 2 (AoD Constant Tone Extension with 2 µs slots). The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in...
Table 4.79. While transmitting the Constant Tone Extension field, the Lower Tester switches antennae using 2 µs slots.

5. The Upper Tester receives an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the LL_CTE_RSP PDU was received, and RSSI shall be set to a valid value. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots). The connEventCounter shall be set to the connection event counter value of the LL_CTE_RSP PDU. Sample_Count shall be set to 8 + (8 x CTETime - 12) / 4. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.

6. Repeat steps 1–5 for each round shown in Table 4.79.

7. If the IUT supports 1 µs slots, repeat steps 1–6, except that in step 2 Requested_CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 µs slots), that in step 3 CTETypeReq shall be set to 1 (AoD Constant Tone Extension with 1 µs slots), that in step 4 CTEType is set to 1 (AoD Constant Tone Extension with 1 µs slots) and the Lower Tester switches antennae using 1 µs slots, and that in step 5 the CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 µs slots) and Sample_Count shall be set to 8 + (8 x CTETime - 12) / 2.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6.23.1 LL/CON/MAS/BV-63-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD – LE 1M PHY]</td>
<td>Master</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.23.2 LL/CON/SLA/BV-67-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD – LE 1M PHY]</td>
<td>Slave</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.23.3 LL/CON/MAS/BV-120-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD – LE 2M PHY]</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.3.6.23.4 LL/CON/SLA/BV-123-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD – LE 2M PHY]</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
</tbody>
</table>

Table 4.80: Constant Tone Extension Request Procedure, IUT Initiated, AoD Test Cases

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT generates an HCI_LE_Connection_IQ_Report event.

4.3.6.24 [Constant Tone Extension Request Procedure, IUT Responding, AoD]

• Test Purpose

Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains a connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT’s response for proper formatting.
• Reference
[10] 2.5, 5.1.12

• Test Case IDs

LL/CON/SLA/BV-68-C
LL/CON/MAS/BV-64-C

• Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.

• Test Procedure

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max-Length_Switching_Pattern between 0x02 and 0x4B and a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_Length_Switching_Pattern and Max_CTE_Length for future use.
For each round from 1–3 based on Table 4.81:

<table>
<thead>
<tr>
<th>Round</th>
<th>Requested_CTE_Length (Step 2, 5-6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>Max_CTE_Length</td>
</tr>
</tbody>
</table>

Table 4.81: Parameter values for each case variation

2. If the CTETime listed in Table 4.81 for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 8.

3. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Transmit_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. CTE_Types shall be set to Allow AoD CTE Response with 2 µs slots (00000100b).

4. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).

5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to the Requested_CTE_Length value specified in Table 4.81. CTETypeReq is set to 2 (AoD Constant Tone Extension with 2 µs slots).

6. The Lower Tester receives an LL_CTE_RSP PDU. The LL_CTE_RSP PDU shall contain the CTEInfo field, with CTETime set to a value greater than or equal to the MinCTELenReq value from step 5, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 µs slots). The packet containing an LL_CTE_RSP PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime.

7. The Lower Tester sends an Empty PDU to the IUT not acknowledging the LL_CTE_RSP PDU.

8. The Lower Tester receives an additional LL_CTE_RSP PDU with the same fields as in step 6.

9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).

10. Repeat steps 1–9 for each round shown in Table 4.81.

11. If the IUT supports 1 µs slots, repeat steps 1–10, except that in step 3 CTE_Types shall be set to Allow AoD CTE Response with 1 µs slots (00000010b), that in step 5 the CTETypeReq shall be set to 1 (AoD Constant Tone Extension with 1 µs slots), and that in step 6 CTEType shall be set to 1 (AoD Constant Tone Extension with 1 µs slots).

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The Lower Tester receives the CTE Response.
4.3.6.25 Unrequested Constant Tone Extension, IUT Receiving, AoA

- **Test Purpose**
  Tests that an IUT is process an unrequested AoA Constant Tone Extension from a remote Lower Tester. Test that the IUT sends the appropriate event to the Upper Tester based on the Lower Tester’s Constant Tone Extension.
  
The Lower Tester maintains a connection and transmits an AoA Constant Tone Extension on a data packet. The Upper Tester observes the event returned by the IUT.

- **Reference**
  [10] 2.5, 5.1.12

- **Initial Condition**
  Connected role as specified in Table 4.83 (PHY as specified in Table 4.83, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
  
The antenna count is defined by the TSPX_number_of_antennae IXIT entry.
For each round from 1 to 3 based on Table 4.82:

<table>
<thead>
<tr>
<th>Round</th>
<th>CTETime (Step 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x02</td>
</tr>
<tr>
<td>2</td>
<td>0x0A</td>
</tr>
<tr>
<td>3</td>
<td>0x14</td>
</tr>
</tbody>
</table>

Table 4.82: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to 0x02. Antenna_IDs[0] and Antenna_IDs[1] shall be
set to the pattern 0, 1. Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).

2. The Lower Tester sends a data channel PDU to the IUT. The data channel PDU contains the CTEInfo field, with CTETime set to the value specified in Table 4.82, RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing the data channel PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the specified CTETime.

3. The Upper Tester receives an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the data channel PDU was received, and RSSI shall be set to a valid value. RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array from step 1. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). The connEventCounter shall be set to the connection event counter value of the data packet PDU.

4. Repeat steps 2–3 for each round shown in Table 4.82.

5. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x00 (disable).

6. Repeat step 2.

7. The Lower Tester expects the IUT to maintain the connection.

8. The Upper Tester does not receive an HCI_LE_Connection_IQ_Report event from the IUT.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6.25.1 LL/CON/MAS/BV-65-C [Unrequested Constant Tone Extension, IUT Receiving, AoA – LE 1M PHY]</td>
<td>Master</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.25.2 LL/CON/SLA/BV-69-C [Unrequested Constant Tone Extension, IUT Receiving, AoA – LE 1M PHY]</td>
<td>Slave</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.3.6.25.3 LL/CON/MAS/BV-121-C [Unrequested Constant Tone Extension, IUT Receiving, AoA – LE 2M PHY]</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.3.6.25.4 LL/CON/SLA/BV-124-C [Unrequested Constant Tone Extension, IUT Receiving, AoA – LE 2M PHY]</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
</tbody>
</table>

Table 4.83: Unrequested Constant Tone Extension, IUT Receiving, AoA Test Cases

• Expected Outcome

Pass Verdict

For all rounds described in the test procedure, the following condition shall occur:

- The IUT generates an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x01.
- The IUT does not generate an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x00.
The IUT acknowledges the Data Channel PDUs containing a Constant Tone Extension sent by the Lower Tester.

### 4.3.6.26 Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection

- **Test Purpose**
  Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when the connection is encrypted. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

  The Lower Tester maintains an encrypted connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- **Reference**
  [10] 2.5, 5.1.12

- **Initial Condition**
  Connected role as specified in Table 4.84 (PHY as specified in Table 4.84, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

  The antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- **Test Procedure**
  Execute the steps in the test case as specified in Table 4.84.

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
</table>
### Test Case

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-125-C</td>
<td>Slave</td>
<td>LE 2M PHY</td>
<td>LL/CON/SLA/BV-122-C</td>
</tr>
</tbody>
</table>

#### Table 4.84: Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection Test Cases

- **Expected Outcome**

  **Pass Verdict**

  Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoA].

- **Test Purpose**

  Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester on an encrypted connection when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

  The Lower Tester maintains an encrypted connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT’s response for proper formatting.

- **Reference**

  [10] 2.5, 5.1.12

- **Test Case IDs**

  LL/CON/SLA/BV-72-C
  LL/CON/MAS/BV-68-C

- **Initial Condition**

  MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- **Test Procedure**

  Same as [Constant Tone Extension Request Procedure, IUT Responding, AoA].

- **Expected Outcome**

  **Pass Verdict**

  Same as [Constant Tone Extension Request Procedure, IUT Responding, AoA].
4.3.6.28 Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection

- Test Purpose
  Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when the connection is encrypted. Test that the IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.
  The Lower Tester maintains an encrypted connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference
  [10] 2.5, 5.1.12

- Initial Condition
  Connected role as specified in Table 4.85 (PHY as specified in Table 4.85, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure
  Execute the steps in the test case as specified in Table 4.85.

- Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6.28.1</td>
<td>Master</td>
<td>LE 1M PHY</td>
<td>LL/CON/MAS/BV-69-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection – LE 1M PHY]</td>
</tr>
<tr>
<td>4.3.6.28.2</td>
<td>Slave</td>
<td>LE 1M PHY</td>
<td>LL/CON/SLA/BV-73-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection – LE 1M PHY]</td>
</tr>
<tr>
<td>4.3.6.28.3</td>
<td>Master</td>
<td>LE 2M PHY</td>
<td>LL/CON/MAS/BV-123-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection – LE 2M PHY]</td>
</tr>
<tr>
<td>4.3.6.28.4</td>
<td>Slave</td>
<td>LE 2M PHY</td>
<td>LL/CON/SLA/BV-126-C [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection – LE 2M PHY]</td>
</tr>
</tbody>
</table>

Table 4.85: Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection Test Cases
• Expected Outcome

Pass Verdict
Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoD].

4.3.6.29 [Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]

• Test Purpose
Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester on an encrypted connection when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains an encrypted connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT’s response for proper formatting.

• Reference
[10] 2.5, 5.1.12

• Test Case IDs
LL/CON/SLA/BV-74-C
LL/CON/MAS/BV-70-C

• Initial Condition
MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
The IUT’s antenna count is defined by the TSPX_number_of_antennae IXIT entry.

• Test Procedure
Same as [Constant Tone Extension Request Procedure, IUT Responding, AoD].

• Expected Outcome
Pass Verdict
Same as [Constant Tone Extension Request Procedure, IUT Responding, AoD].

4.3.6.30 Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC

• Test Purpose
Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains a connection and has been configured to transmit Connection CTE Responses with incorrect CRCs. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

• Reference
[10] 2.5, 5.1.12
• Initial Condition

Connected role as specified in Table 4.86 (PHY as specified in Table 4.86, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The antenna count is defined by the TSPX_number_of_antennae IXIT entry.

• Test Procedure

Execute the steps in the test case as specified in Table 4.86, except that in step 5, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC, and that in step 6, the Packet_Status shall be set to 0x01 or 0x02.

Repeat step 5, so that the Lower Tester transmits the packet containing the Constant Tone Extension field using a valid CRC, and in step 6, the Packet_Status shall be set to 0x00.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-75-C</td>
<td>Slave</td>
<td>LE 1M PHY</td>
<td>LL/CON/SLA/BV-60-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 1M PHY]</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-124-C</td>
<td>Master</td>
<td>LE 2M PHY</td>
<td>LL/CON/MAS/BV-119-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 2M PHY]</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-127-C</td>
<td>Slave</td>
<td>LE 2M PHY</td>
<td>LL/CON/SLA/BV-122-C [Constant Tone Extension Request Procedure, IUT Initiated, AoA – LE 2M PHY]</td>
</tr>
</tbody>
</table>

Table 4.86: Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC Test Cases

• Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connection_IQ_Report events with Packet_Status set to indicate the CRC was incorrect when receiving a packet containing the Constant Tone Extension field with incorrect CRC.
Inconclusive Verdict

The IUT does not generate any HCI_LE_Connection_IQ_Report events when receiving a packet containing the Constant Tone Extension field with incorrect CRC.

• Fail Verdict

The IUT does not generate an HCI_LE_Connection_IQ_Report events with Packet_Status set to 0x00 (CRC was correct) when receiving a packet containing the Constant Tone Extension field with a valid CRC.

4.3.6.31  Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC

• Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains a connection and has been configured to transmit Connection CTE Responses with incorrect CRCs. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

• Reference

[10] 2.5, 5.1.12

• Initial Condition

Connected role as specified in Table 4.87 (PHY as specified in Table 4.87, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

• Test Procedure

Execute the steps in the test case as specified in Table 4.87, except that in step 4, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC, and that in step 5, the Packet_Status shall be set to 0x01 or 0x02.

Repeat step 4, so that the Lower Tester transmits the packet containing the Constant Tone Extension field using a valid CRC, and in step 5, the Packet_Status shall be set to 0x00.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
</table>
### Test Case

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
<th>PHY</th>
<th>Test Case to Execute</th>
</tr>
</thead>
</table>

Table 4.87: Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC Test Cases

• **Expected Outcome**

**Pass Verdict**

The IUT generates HCI_LE_Connection_IQ_Report events with Packet_Status set to indicate the CRC was incorrect when receiving a packet containing the Constant Tone Extension field with incorrect CRC.

**Inconclusive Verdict**

The IUT does not generate any HCI_LE_Connection_IQ_Report events when receiving a packet containing the Constant Tone Extension field with incorrect CRC.

• **Fail Verdict**

The IUT does not generate an HCI_LE_Connection_IQ_Report events with Packet_Status set to 0x00 (CRC was correct) when receiving a packet containing the Constant Tone Extension field with a valid CRC.

#### 4.3.6.32 Data Length Update – Preserve Parameters After a PHY Change

• **Test Purpose**

Verify that the IUT preserves the data length parameters when transitioning between the LE 1M PHY and the PHY specified in Table 4.88 in the role specified.

• **Reference**

[8] 5.1.9, [10] 4.5.10

• **Initial Condition**

State: Connected in the specified role on the LE 1M PHY. The values for the IUT’s maximum supported TxOctets (supportedMaxTxOctets) and supported maximum TxTime (supportedMaxTxTime) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.
• Test Procedure

Connection Established. IUT Role As Specified.

REPEAT IF NECESSARY

Connection Established. IUT Role As Specified.

NOTE: At no time during this procedure shall the Lower Tester initiate a Data Length Update procedure.

1. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to supportedMaxTxTime and TxOctets set to supportedMaxTxOctets. The Upper Tester receives a successful HCI_Command_Complete in response.

2. The IUT may send an LL_LENGTH_REQ PDU to the Lower Tester.

3. If the Lower Tester receives an LL_LENGTH_REQ PDU from the IUT, it sends an LL_LENGTH_RSP PDU to the IUT in response.

4. If the IUT initiated the Data Length Update procedure in step 2, then the Upper Tester receives an HCI_LE_Data_Length_Change event, and the Upper Tester notes the MaxTxTime, MaxTxOctets, MaxRxTime, and MaxRxOctets values.

5. If a Data Length Update procedure was not initiated, then repeat the previous steps. In the next iteration, set TxTime and TxOctets to intermediate values according to the following, ignoring any fractional portion of the result:
   \[ \text{TxTime} = ((\text{supportedMaxTxTime} - 328) / 2) + 328 \]
   \[ \text{TxOctets} = ((\text{supportedMaxTxOctets} - 27) / 2) + 27 \]
   If the IUT fails to generate a Data Length Update on the second attempt, then this test is inconclusive.

6. The Upper Tester sends an HCI_LE_Set_PHY setting the PHY for both transmit and receive to the PHY specified in Table 4.88 and receives a successful HCI_Command_Status in return.

Figure 4.319: Data Length Update – Preserve Parameters After a PHY Change
7. The Upper Tester receives an LE_PHY_Update_Complete event indicating that one or both PHYs have changed. If an LE_PHY_Update_Complete event is not received or the Tx PHY has not changed, then this test is inconclusive.

8. If the Upper Tester receives an HCI_LE_Data_Length_Change event, then this test fails unless the PHY in step 6 was the LE Coded PHY and the only change is to increase MaxTxTime, MaxRxTime, or both to 2704.

9. The Upper Tester sends an HCI_LE_Set_PHY setting the PHY for both transmit and receive to the LE 1M PHY and receives a successful HCI_Command_Status in return.

10. The Upper Tester receives an LE_PHY_Update_Complete event indicating that one or both PHYs have changed. If an LE_PHY_Update_Complete event is not received or the Tx PHY has not changed, then this test is inconclusive.

11. If the Upper Tester receives an HCI_LE_Data_Length_Change event, then this test fails unless the PHY in step 9 was the LE Coded PHY and the only change is to revert MaxTxTime, MaxRxTime, or both to the values noted in step 4.

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-129-C</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>LL/CON/SLA/BV-130-C</td>
<td>Slave</td>
<td>LE Coded PHY</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-126-C</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-127-C</td>
<td>Master</td>
<td>LE Coded PHY</td>
</tr>
</tbody>
</table>

*Table 4.88: Master Data Length Update – Reset on Disconnect Test Cases*

**Pass Verdict**

- In step 8, the Upper Tester did not receive an HCI_LE_Data_Length_Change event, unless the PHY in step 6 was the LE Coded PHY and the only change was to increase MaxTxTime, MaxRxTime, or both to 2704.

- In step 11, the Upper Tester did not receive an HCI_LE_Data_Length_Change event, unless the PHY in step 9 was the LE Coded PHY and the only change was to revert MaxTxTime, MaxRxTime, or both to the values noted in step 4.

**Fail Verdict**

- In step 8, the Upper Tester received an HCI_LE_Data_Length_Change event, unless the PHY in step 6 was the LE Coded PHY and the only change was to increase MaxTxTime, MaxRxTime, or both to 2704.

- In step 11, the Upper Tester received an HCI_LE_Data_Length_Change event, unless the PHY in step 9 was the LE Coded PHY and the only change was to revert MaxTxTime, MaxRxTime, or both to the values noted in step 4.

**Inconclusive Verdict**

The IUT does not initiate the Data Length Update procedure when the Upper Tester sends the HCI_LE_Set_Data_Length commands.

The Tx PHY does not change when requested.
4.3.6.33 Data Length Update – Retransmission During an Update

• Test Purpose

Verify that the IUT acting in the role specified in Table 4.89 does not refragment a packet waiting for retransmission even when a data length update is received and the new parameters forbid the packet.

• Reference

[8] 5.1.9, [10] 4.5.10

• Initial Condition

State: Connected in the specified role. Values for maximum TxOctets supported (supportedMaxTxOctets) and TxTime (supportedMaxTxTime) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in “Time”

The values of these parameters will depend on the features supported by the IUT.
**Test Procedure**

1. Upper Tester commands IUT to set TxOctets and TxTime to supportedMaxTxOctets and supportedMaxTxTime, respectively, via an HCI_LE_Set_Data_Length command. The Upper Tester receives a successful HCI_Command_Complete in response.
2. The IUT may send an LL_LENGTH_REQ to Lower Tester with new values.
3. If the Lower Tester receives an LL_LENGTH_REQ PDU from the IUT, it sends an LL_LENGTH_RSP to the IUT accepting the values. It notes the IUT’s MaxTxOctets.
4. If the IUT initiated a Data Length Update procedure in step 2, then the IUT sends the Upper Tester an HCI_LE_Data_Length_Change event indicating values have been changed.
5. If a Data Length Update procedure was not initiated, then repeat the previous steps. In the next iteration, set TxTime and TxOctets to intermediate values according to the following, ignoring any fractional portion of the result:
   
   \[
   \text{TxTime} = \left(\frac{\text{supportedMaxTxTime} - 328}{2}\right) + 328 \\
   \text{TxOctets} = \left(\frac{\text{supportedMaxTxOctets} - 27}{2}\right) + 27
   \]
If the IUT fails to generate a Data Length Update on the second attempt, then this test is inconclusive.

6. The Upper Tester sends a stream of ACL data packets that are supportedMaxTxOctets in size.
7. The Lower Tester begins receiving data packets that are the IUT’s MaxTxOctets in size.
8. The Lower Tester refuses to acknowledge one of the data packets.
9. The Lower Tester immediately sends an LL_LENGTH_REQ specifying connMinOctetsLimitSpec and connMinTimeLimitSpec as the max octet and max time values, respectively, for both Rx and TX.
10. After 5 connection intervals, the Lower Tester acknowledges the last data packet.
11. The Lower Tester receives the previous data packet. The length remains the IUT’s MaxTxOctets in size.

• Expected Outcome

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-131-C</td>
<td>Slave</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-128-C</td>
<td>Master</td>
</tr>
</tbody>
</table>

*Table 4.89: Data Length Update – Retransmission During an Update*

**Pass Verdict**

After the delay, the Lower Tester receives the previous data packet and it is the IUT’s MaxTxOctets in size.

**Inconclusive Verdict**

The IUT does not initiate the Data Length Update procedure when the Upper Tester sends the HCI_LE_Set_Data_Length commands.

**4.3.6.34 Data Length Update – Handling Invalid Data Length Responses**

• Test Purpose

Verify that IUT role as specified in Table 4.91 correctly rejects reception of an LL_LENGTH_RSP PDU with invalid values on a PHY specified in Table 4.91 and continues to receive data packets from the Lower Tester.

• Reference

[8] 5.1.9, [10] 4.5.10

• Initial Condition

State: Connected in the specified role using the specified PHY. Values for maximum TxOctets supported (supportedMaxTxOctets), TxTime (supportedMaxTxTime), RxOctets (supportedMaxRxOctets) and RxTime (supportedMaxRxTime) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in “Octet”
- \( \text{connMinTimeLimitSpec} \) = the minimum permitted value for parameters with names ending in "Time"
- \( \text{connMaxTimeLimitSpec} \) = the maximum permitted value for parameters with names ending in "Time"

The values of these parameters will depend on the features supported by the IUT.

• Test Procedure

![Diagram showing test procedure]

**Figure 4.321: Data Length Update – Handling Invalid Data Length Responses**

1. For each round in Table 4.90, perform steps 2–7.
2. The Upper Tester sends an HCI\_LE\_Set\_Data\_Length command with TxOctets set to a random value from \( \text{connMinOctetsLimitSpec} \) to \( \text{supportedMaxTxOctets} \), and a TxTime set to a random value from \( \text{connMinTimeLimitSpec} \) to \( \text{supportedMaxTxTime} \). The Upper Tester receives a successful HCI\_Command\_Complete in response.
3. The IUT may send an LL\_LENGTH\_REQ to the IUT. If the IUT does not, retry step 2 with new random values up to 10 additional times. If after 10 attempts the IUT has not issued an LL\_LENGTH\_REQ to the Lower Tester, the test is inconclusive.
4. The Lower Tester responds with an LL\_LENGTH\_RSP with the values specified in Table 4.90.
5. The Upper Tester may receive an HCI\_LE\_Data\_Length\_Change event, such as if the IUT truncates the invalid value to the nearest valid value. If the IUT reports any invalid values, the test fails.
6. The Lower Tester sends PDU data packets of the size specified by the IUT when the Lower Tester’s MaxTxOctets was invalid, and of size MaxTxOctets when valid.
7. The IUT sends the data packets to the Upper Tester.

<table>
<thead>
<tr>
<th>Round</th>
<th>LL_LENGTH_RSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MaxTxOctets</td>
</tr>
<tr>
<td>1</td>
<td>( \text{connMinOctetsLimitSpec} - 1 )</td>
</tr>
<tr>
<td>2</td>
<td>( \text{connMinOctetsLimitSpec} )</td>
</tr>
</tbody>
</table>
Link Layer (LL) / Test Suite

<table>
<thead>
<tr>
<th>Round</th>
<th>LL_LENGTH_RSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MaxTxOctets</td>
</tr>
<tr>
<td>3</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>4</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>5</td>
<td>supportedMaxRxOctets + 1</td>
</tr>
<tr>
<td>6</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>7</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>8</td>
<td>connMaxOctetsLimitSpec + 1</td>
</tr>
<tr>
<td>9</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>10</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>11</td>
<td>connMinOctetsLimitSpec</td>
</tr>
<tr>
<td>12</td>
<td>connMinOctetsLimitSpec</td>
</tr>
</tbody>
</table>

NOTE: Invalid value entry in italics

Table 4.90: LL_LENGTH_RSP content

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BI-10-C</td>
<td>Slave</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>LL/CON/SLA/BI-11-C</td>
<td>Slave</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>LL/CON/SLA/BI-12-C</td>
<td>Slave</td>
<td>LE Coded PHY</td>
</tr>
<tr>
<td>LL/CON/MAS/BI-07-C</td>
<td>Master</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>LL/CON/MAS/BI-08-C</td>
<td>Master</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>LL/CON/MAS/BI-09-C</td>
<td>Master</td>
<td>LE Coded PHY</td>
</tr>
</tbody>
</table>

Table 4.91: Master Data Length Update – Reset on Disconnect Test Cases

**Pass Verdict**

- In step 5, the Upper Tester does not receive an HCI_LE_Data_Change event, or it does receive an HCI_LE_Data_Change event but none of the received values are invalid.
- The test procedure is executed successfully, and the IUT is able to receive data packets from the Lower Tester.

**Inconclusive Verdict**

The IUT does not issue an LL_LENGTH_REQ.
4.3.6.35 Data Length Update – Peer Does Not Support LE Coded PHY

• Test Purpose
  Verify that if the IUT’s peer device does not support the LE Coded PHY feature, then the MaxRxTime and MaxTxTime fields in the LL_LENGTH_REQ and LL_LENGTH_RSP PDUs shall be set to a value less than or equal to 2120 microseconds.

• Reference
  [8] 5.1.9, [10] 4.5.10

• Initial Condition
  State: Connected in the specified role on the LE 1M PHY. The value for the IUT’s supported maximum TxTime (supportedMaxTxTime) has been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.
  Note: in this test, the following term is used to refer to the minima in Volume 6 Part B Table 4.3 of the Core Specification:
  - connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
  The Lower Tester is configured to not support the LE Coded PHY.
  The IUT and the Lower Tester have not executed a Feature Exchange procedure.
Test Procedure

- The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with \( T_x\text{Time} \) set to \( \text{supportedMaxTxTime} \) and \( T_x\text{Octets} \) set to 27. The Upper Tester receives a successful HCI_Command_Complete in response.
- The IUT may issue an LL_LENGTH_REQ to the Lower Tester. If \( \text{MaxRxTime} \) or \( \text{MaxTxTime} \) are greater than 2120, then the test fails. The Lower Tester responds with an LL_LENGTH_RSP.
- If the IUT initiated a Data Length Update procedure in step 2, then the Upper Tester receives an HCI_LE_Data_Length_Changed event. If \( \text{MaxTxTime} \) or \( \text{MaxRxTime} \) are greater than 2120, then the test fails.
- If an LL_LENGTH_REQ was not sent to the Lower Tester, then return to step 1 and increase \( T_x\text{Octets} \) by 20. Repeat until the \( T_x\text{Octets} \) value would exceed \( \text{connMinOctetsLimitSpec} \) or 10.

Figure 4.322: Data Length Update – Peer Does Not Support LE Coded PHY
rounds, whichever comes first. If that limit is reached and no LL_LENGTH_REQ has been issued, then the test is inconclusive.

5. The Lower Tester sends PDU data packets of the size agreed upon in the Data Length Update Procedure.

6. The IUT sends the data packets to the Upper Tester.

7. If the “Requesting Feature Setup” feature is not supported for the respective role, then the test is complete. If the feature is supported, continue to the following steps.

8. The Upper Tester sends an HCI_LE_Read_Remote_Functions command to the IUT using the Connection_Handle for the connection to the Lower Tester and receives a successful HCI_Command_Status event in response.

9. If the IUT is in the Master role, it sends an LL_FEATURE_REQ to the Lower Tester. If the IUT is in the Slave role, it sends an LL_SLAVE_FEATURE_REQ.

10. The Lower Tester responds with an LL_FEATURE_RSP indicating it does not support the LE Coded PHY.

11. The Upper Tester receives an HCI_LE_Read_Remote_Functions_Complete event indicating that the Lower Tester does not support the LE Coded PHY.

12. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to supportedMaxTxTime and TxOctets set to 27. The Upper Tester receives a successful HCI_Command_Complete in response.

13. The IUT may issue an LL_LENGTH_REQ to the Lower Tester. If MaxRxTime or MaxTxTime are greater than 2120, then the test fails. The Lower Tester then responds with an LL_LENGTH_RSP.

14. If the IUT initiated a Data Length Update procedure in step 13, then the Upper Tester does not receive an HCI_LE_Data_Length_Changed event. If it does receive one, then the test fails.

15. If an LL_LENGTH_REQ was not sent to the Lower Tester, then return to step 12 and increase TxOctets by 20. Repeat until the TxOctets value would exceed connMinOctetsLimitSpec or 10 rounds, whichever comes first. If that limit is reached and no LL_LENGTH_REQ has been issued, then the test is concluded.

16. The Lower Tester sends PDU data packets of the size agreed upon in the Data Length Update Procedure.

17. The IUT sends the data packets to the Upper Tester.

• Expected Outcome

<table>
<thead>
<tr>
<th>Test Case ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CON/SLA/BV-132-C</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-129-C</td>
</tr>
</tbody>
</table>

Table 4.92: Data Length Update – Peer Does Not Support LE Coded PHY

Pass Verdict
MaxRxTime or MaxTxTime are less than or equal to 2120.
The Upper Tester receives data packets from the IUT.

Inconclusive Verdict
The IUT does not issue an LL_LENGTH_REQ to the Lower Tester.
4.3.6.36  Constant Tone Extension Disabled on PHY Change

- Test Purpose
  Tests that an IUT stops sending CTE requests after changing to a PHY that doesn’t allow Constant Tone Extensions.
  The Upper Tester configures the IUT for CTE Requests, initiates periodic CTE Requests, and observes the responses returned by the IUT. The Lower Tester requests a switch to a PHY that doesn’t allow Constant Tone Extensions.

- Reference
  [10] 2.5, 5.1.12

- Test Case IDs
  LL/CON/SLA/BV-134-C
  LL/CON/MAS/BV-131-C

- Initial Condition
  MAS/SLA: Connected (LE 1M PHY, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).
Test Procedure

Figure 4.323: Constant Tone Extension Request Procedure, IUT Initiated, Periodic, Part 1
Repeat for LE Coded PHY and LE 1M PHY
LE_PHY_REQ
(TX_PHYS, RX_PHYS=0x07)
LE_PHY_RSP
(TX_PHYS)
LE_PHY_UPDATE_IND
IUT: Slave
IUT: Master
Optional
LE_PHY_REQ
(TX_PHYS)
LE_PHY_RSP
(TX_PHYS)
LE_PHY_UPDATE_IND
IUT: Master
IUT: Slave
LE_PHY_UPDATE_IND
4 *
CTE_Request_Interval
LE_CTE_REQ
X
HCI_LE_Connection_IQ_Reports
X
Figure 4.324: Constant Tone Extension Request Procedure, IUT Initiated, Periodic, Part 2

1. The Upper Tester sends an HCI_LE_Set_PHY command with ALL_PHYs set to 3 and receives an HCI_Command_Status event in reply. If the IUT initiates the PHY Update Procedure, the Lower Tester shall take part in such a way as to leave the connection using the LE 1M PHY in both directions.
2. The IUT sends an LE_PHY_Update_Complete event to the Upper Tester showing that the connection is using the LE 1M PHY in both directions.
3. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 0, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
4. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x10. Requested_CTE_Length shall be set to 0x02. Requested_CTE_Type shall be set to a valid value for a CTE type that the IUT supports.
5. The IUT sends an LL_CTE_REQ PDU to the Lower Tester.
6. The Lower Tester sends an LL_CTE_RSP PDU. The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field of length matching the Requested_CTE_Length specified in step 4.

7. The IUT sends an HCI_LE_Connection_IQ_Report event to the Upper Tester.

8. Repeat steps 5–7 until 5 Constant Tone Extensions have been transmitted. The Lower Tester receives each successive LL_CTE_RSQ PDU after a number of connection events equal to or greater than the CTE_Request_Interval value in step 4 since receiving the last such PDU. These steps will continue to happen in parallel with the following steps until the Instant of the PHY Update Procedure is reached.

9. The Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY change to the LE Coded PHY. TX_PHYS shall be set to only specify the LE Coded PHY; RX_PHYS shall allow all PHYs.

10. The Lower Tester and IUT complete the PHY Update Procedure. If the Lower Tester is now transmitting on the LE Coded PHY, skip to step 13.

11. The Upper Tester sends an HCI_LE_Set_PHY command with ALL_PHYs set to 1 and RX_PHYs set to only prefer the LE Coded PHY and receives an HCI_Command_Status event in reply.

12. The IUT initiates a PHY Update Procedure with the Lower Tester that allows the Lower Tester to transmit on the LE Coded PHY. The Lower Tester and IUT complete the procedure. If the Lower Tester transmit PHY is not the LE Coded PHY, the test ends with an Inconclusive Verdict.

13. The Upper Tester receives an LE_PHY_Update_Complete event showing that the RX_PHY is the LE Coded PHY.

14. After the Instant where the Lower Tester switches to transmitting on the LE Coded PHY, wait for at least 4 times the number of connection events specified at CTE_Request_Interval in step 4. During this time, the Lower Tester receives no LL_CTE_REQ PDUs and the Upper Tester receives no HCI_LE_Connection_IQ_Report events.

15. Repeat steps 9–14, except that “LE Coded PHY” is replaced by “LE 1M PHY” throughout.

16. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to 0x00 (Disable). The IUT responds with a successful HCI_Command_Complete event.

• Expected Outcome

Pass Verdict

The IUT periodically requests Constant Tone Extensions, with at least CTE_Request_Interval connection events between requests.

The IUT generates HCI_LE_Connection_IQ_Report events.

The Lower Tester verifies that it stops receiving LL_CTE_REQ PDUs after the PHY changes to LE Coded PHY.

The Lower Tester verifies that it doesn’t receive LL_CTE_REQ PDUs after the PHY changes back to LE 1M PHY.

Inconclusive Verdict

The IUT does not change its receive PHY to the LE Coded PHY in step 12 or to the LE 1M PHY in step 15.
4.4 TIM
Tests that the IUT behaves according to timing requirements in procedures where there may be variable timing. The timing requirements apply for event intervals when the IUT is in low power mode and for packet timing and intervals when the IUT is in active mode.

4.4.1 ADV
Tests that the IUT behaves according to timing constraints as an advertiser.

4.4.1.1 LL/TIM/ADV/BV-01-C [Earliest Transmission to Advertiser]
• Test Purpose
  Test that an advertiser IUT responds to a scan request sent using the minimum timing between packets (T_IFS – 2 µsec).
  The Lower Tester requests information from the IUT and receives a response.

• Reference
  [3] 4.2, 4.4.2.3

• Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name)
**Test Procedure**

1. Configure Lower Tester to start active scanning.
2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
3. Lower Tester receives an ADV_IND packet from the IUT and responds with a SCAN_REQ packet, using the minimum time after the end of the advertising packet (T_IFS – 1.5 μsec).
4. Lower Tester receives a SCAN_RSP packet from the IUT T_IFS after the SCAN_REQ.
5. Repeat steps 3–4 100 times.

**Expected Outcome**

**Pass Verdict**

The IUT responds at least to 95 percent of the SCAN_REQ packets sent by the Lower Tester in step 3.

**Notes**

Lower Tester is configured to use 1.5μsec instead of 2μsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.
4.4.1.2  LL/TIM/ADV/BV-02-C [Latest Transmission to Advertiser]

- **Test Purpose**
  Test that an advertiser IUT responds to a scan request sent using the maximum timing between packets (T_IFS + 2 µsec).
  The Lower Tester requests information from the IUT and receives a response.

- **Reference**
  [3] 4.2, 4.4.2.3

- **Initial Condition**
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Undirected Advertising (selected Adv.Interval.Min, selected Adv.Interval.Max, supported type of own address, selected advertising channels, Length of device name used, common device name)

- **Test Procedure**

  1. Configure Lower Tester to start active scanning.
  2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.

*Figure 4.326: LL/TIM/ADV/BV-02-C [Latest transmission to Advertiser]*
3. Lower Tester receives an ADV_IND packet from the IUT and responds with a SCAN_REQ packet, using the maximum time after the end of the advertising packet (T_IFS + 1.5 μsec).
4. Lower Tester receives a SCAN_RSP packet from the IUT T_IFS after the SCAN_REQ.
5. Repeat steps 3–4 100 times.

• Expected Outcome

Pass Verdict

The IUT responds at least to 95 percent of the SCAN_REQ packets sent by the Lower Tester in step 3.

• Notes

Lower Tester is configured to use 1.5μsec instead of 2μsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.1.3 Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser

• Test Purpose

Tests that an advertiser IUT sends scannable undirected ADV_EXT_IND with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel. Tests that an advertiser IUT responds to a scan request on the secondary channel when scanner uses the minimum transmission time to advertiser.

• Reference

[10] 4.4.2.5.4

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
The Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter bit 1 (Scannable Advertising) shall be set and all other bits cleared. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set as specified in Table 4.93.

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Response_Data command to the IUT containing at least 1 octet of Scan Response Data.

3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

4. The Lower Tester receives an ADV_EXT_IND PDU with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent T_MAFS after the ADV_EXT_IND PDU or later. The Lower Tester sends an AUX_SCAN_REQ PDU the minimum time after the end of the AUX_ADV_IND PDU (T_IFS – 1.5 μsec).

5. Lower Tester receives an AUX_SCAN_RSP packet from the IUT T_IFS (plus or minus 2 μsecs) after the AUX_SCAN_REQ.

6. Repeat steps 4–5 100 times.

Figure 4.327: Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser
7. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

- **Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Advertising PHY</td>
<td>Secondary Advertising PHY</td>
</tr>
<tr>
<td>4.4.1.3.1 LL/TIM/ADV/BV-03-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.4.1.3.2 LL/TIM/ADV/BV-05-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser – LE 2M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td>4.4.1.3.3 LL/TIM/ADV/BV-06-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser – LE Coded PHY]</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

Table 4.93: Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser Test Cases

- **Expected Outcome**

  **Pass Verdict**

  The IUT responds to at least 95 percent of the AUX_SCAN_REQ packets sent by the Lower Tester in step 4 within T_IFS (plus or minus 2 µsecs).

  The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.

- **Notes**

  Lower Tester is configured to use 1.5µsec instead of 2µsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.1.4 **Extended Advertising, Secondary Channel, Latest Transmission to Advertiser**

- **Test Purpose**

  Tests that an advertiser IUT sends scannable undirected ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel. Tests that an advertiser IUT responds to a scan request on the secondary channel when scanner uses the maximum transmission time to advertiser (T_IFS + 2 µsec).

- **Reference**

  [10] 4.4.2.5.4

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter bit 1 (Scannable Advertising) shall be set and all other bits cleared. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set as specified in Table 4.94.

2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Response_Data command to the IUT containing at least 1 octet of Scan Response Data.

3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

4. Lower Tester receives an ADV_EXT_IND PDU with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent T_MAFS after the
ADV_EXT_IND PDU or later. The Lower Tester sends an AUX_SCAN_REQ PDU the maximum time after the end of the AUX_ADV_IND PDU \((T_{IFS} + 1.5 \ \mu \text{sec})\).

5. Lower Tester receives an AUX_SCAN_RSP PDU from the IUT \(T_{IFS}\) after the AUX_SCAN_REQ.

6. Repeat steps 4–5 100 times.

7. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

**Test Case Configuration**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Advertising PHY</strong></td>
<td><strong>Secondary Advertising PHY</strong></td>
</tr>
<tr>
<td><strong>4.4.1.4.1</strong> LL/TIM/ADV/BV-04-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser – LE 1M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.4.1.4.2</strong> LL/TIM/ADV/BV-07-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser – LE 2M PHY]</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.4.1.4.3</strong> LL/TIM/ADV/BV-08-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser – LE Coded PHY]</td>
<td>0x03 (LE Coded PHY)</td>
</tr>
</tbody>
</table>

*Table 4.94: Extended Advertising, Secondary Channel, Latest Transmission to Advertiser Test Cases*

**Expected Outcome**

**Pass Verdict**

The IUT responds to at least 95 percent of the AUX_SCAN_REQ packets sent by the Lower Tester in step 4.

The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to \(T_{MAFS}\).

**Notes**

Lower Tester is configured to use 1.5 \(\mu\)sec instead of 2 \(\mu\)sec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

**4.4.2 SLA**

Tests that the IUT behaves according to timing constraints in the slave role. The slave timing requirements apply both in low power and active mode.

**4.4.2.1 LL/TIM/SLA/BV-01-C [Adjusting Anchor Point]**

**Test Purpose**

Test that a slave IUT on accepting a parameter update from the master, adopts a new anchor point when starting to use the new parameters.
The Lower Tester acts in the master role in the connection, sending a connection parameter update packet to the IUT until it accepts it, then takes the new parameters into use varying the time to the earliest and latest possible. The Lower Tester observes the slave transmissions on the data channels used.

- Reference
  [3] 5.1.1

- Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, 200 ms timeout, common channel map, any SCA value)

- Test Procedure
  Execute the test procedure using a connection supervision timer parameter of 200 ms and zero slave latency. Update using the same connection parameters.
1. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum Transmission Window Size and to the minimum Transmission Offset and receives a packet from the IUT acknowledging the connection update request.

2. Lower Tester sends an empty DATA packet to the IUT and receives a response in all events until the event count matches the indicated time of connection update.
3. Send an empty LL DATA packet starting the event at connection interval, using the common data channel selection parameters. Receive an empty LL DATA packet in response from the IUT, T_IFS after the data packet sent on the same data channel. Repeat up to 15 times.

4. Interleave with step 3: Receive no HCI_LE_Connection_Update_Complete event from the IUT.

5. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum Transmission Window Size and to the maximum Transmission Offset and receives a packet from the IUT acknowledging the connection update request.

6. Repeat step 2.

7. Repeat steps 3–4.

- **Test Condition**

  The parameters in this test are calculated for a BER of 0.1 percent or better.

- **Expected Outcome**

  **Pass Verdict**

  The test procedure executes successfully, with the IUT adopting the new anchor point and maintaining the connection.

  The IUT produces no HCI event reporting the connection update.

- **Notes**

  The calculations in the tests 'Earliest/Latest Transmission Start to Slave' can be used to define the connection parameters and test procedure details required. A difference in the parameter application is that the transmission window size is larger and the window offset is used.

**4.4.2.2 LL/TIM/SLA/BV-02-C [Earliest Transmission Start to Slave]**

- **Test Purpose**

  Test that the slave IUT is able to establish and maintain a connection with a master that uses the earliest possible timing for the first transmission.

  The Lower Tester acts first in the initiating state, then times the first transmission in the data channel at the earliest time possible.

- **Reference**

  [3] 4.5.5

- **Initial Condition**

  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name)

- **Test Procedure**

  Execute the test using the common connection interval and data channel selection parameters, a connection supervision timer of 10 s and a zero slave latency in the connection request packet using the minimum SCA for the most deviation by clock drift. Use a minimum transmission window size (1.25 ms) with 0 offset.
1. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel, respond with a CONNECT_IND packet T_IFS after the end of the advertising packet.
2. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT with the parameters sent in step 1.
3. Lower Tester sends an empty LL DATA packet starting the first event at connection interval transmitting at the leading edge of the window, using the common data channel selection parameters. Lower Tester receives an empty LL DATA packet in response from the IUT, T_IFS after the data packet sent on the same data channel. Repeat up to 10 times.
4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 2).

- **Test Condition**
  
The parameters in this test are calculated for a BER of 0.1 percent or better.

- **Expected Outcome**
  
  **Pass Verdict**
  
The test procedure executes successfully, with the IUT starting slave transmissions after the events skipped.

### 4.4.2.3 LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]

- **Test Purpose**
  
  Test that the slave IUT is able to establish a connection with a master that uses the latest possible timing for the first transmission.
The Lower Tester acts first in the initiating state, then times the first transmission in the data channel at the latest time required.

- **Reference**
  
  [3] 4.5.5

- **Initial Condition**
  
  State: Undirected Advertising (selected Adv_INTERVAL_Min, selected Adv_INTERVAL_Max, supported type of own address, selected advertising channels, Length of device name used, common device name)

- **Test Procedure**
  
  Execute the test, using the common connection interval and data channel selection parameters. Use a connection supervision timer of 690 ms, zero slave latency, the minimum SCA and a maximum transmission window and maximum transmission offset in the connection request packet.

1. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
2. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT with the parameters sent in step 1.
3. Lower Tester sends an empty LL DATA packet starting the first event at connection interval transmitting at the trailing edge of the window, using the common data channel selection parameters.
4. Lower Tester receives an empty LL DATA packet in response from the IUT, T_IFS after the data packet sent on the same data channel. Repeat up to 10 times.

*Figure 4.331: LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]*
5. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 2).

- **Test Condition**
  The parameters in this test are calculated for a BER of 0.1 percent or better.

- **Expected Outcome**
  **Pass Verdict**
  The test procedure executes successfully, with the IUT setting up the connection by starting slave transmissions after the skipped events.

### 4.4.2.4 LL/TIM/SLA/BV-04-C [Shortest Connection Interval]

- **Test Purpose**
  Test that a slave IUT is able to maintain a connection with a master using the minimum timing between events (ConnInterval – 16µsec).
  The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the minimum time required between events.

- **Reference**
  [3] 4.2

- **Initial Condition**
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, -20 ppm SCA value)

- **Test Procedure**
  Execute the test using the common data channel selection parameters, a nominal connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

*Figure 4.332: LL/TIM/SLA/BV-04-C [Earliest Transmission to Slave]*
1. Lower Tester sends DATA packets once a connection interval to the IUT is made using the data channel selection. Connection Interval will be set to nominal ConnInterval – 15.5 µsec. Receive an empty DATA packet from the IUT T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Repeat up to 10 times. If the IUT does not respond to one Lower Tester packet, next packet sent by the Lower Tester will be at nominal ConnInterval, according to Figure 4.333.

2. Repeat step 1 five times.

3. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from preamble).

   • Expected Outcome

   **Pass Verdict**

   The test procedure executes successfully, with the IUT maintaining the connection.

   The IUT responds to all 10 packets sent by the Lower Tester in at least 3 of 5 repetitions.

   • Notes

   Lower Tester is configured to use 15.5µsec instead of 16µsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

   The parameters in this test are calculated for a BER of 0.1 percent or better.

4.4.2.5 LL/TIM/SLA/BV-05-C [Longest Connection Interval]

   • Test Purpose

   Test that a slave IUT is able to maintain a connection with a master using the maximum timing between events (ConnInterval + 16 µsec).

   The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the maximum time required between events.

   • Reference

   [3] 4.2
• **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, 20 ppm SCA value).

• **Test Procedure**

Execute the test using the common data channel selection parameters, a nominal connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

**Figure 4.334: LL/TIM/SLA/BV-05-C [Latest Transmission to Slave]**

**Figure 4.335: LL/TIM/SLA/BV-05-C Timing sequence**

1. Lower Tester sends a DATA packet once a connection interval to the IUT, using the using the data channel selection parameters. Connection Interval will be set to nominal ConnInterval + 15.5 μsec. Receive an empty DATA packet from the IUT T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Repeat up to 10 times. If the IUT does not respond to one Lower Tester packet, next packet sent by the Lower Tester will be at nominal ConnInterval, according to Figure 4.335.
2. Repeat step 1 five times.
3. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble.

- Expected Outcome
  
  **Pass Verdict**
  
  The test procedure executes successfully, with the IUT maintaining the connection.
  
  The IUT responds to all 10 packets sent by the Lower Tester in at least 3 of 5 repetitions.

- Notes
  
  Lower Tester is configured to used 15.5µsec instead of 16µsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

  The parameters in this test are calculated for a BER of 0.1 percent or better.

### 4.4.2.6 LL/TIM/SLA/BV-06-C [Earliest Transmissions to Slave]

- Test Purpose
  
  Test that a slave IUT is able to maintain a connection with a master using the minimum timing between packets (T_{IFS} – 2 µsec).

  The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the minimum time required between packets.

- Reference
  
  [3] 4.2

- Initial Condition
  
  State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure
  
  Execute the test using the common data channel selection parameters, a connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.
1. Lower Tester sends a DATA packet to the IUT at the beginning of the Connection Event, setting MD bit to 1. Receive an empty DATA packet from the IUT T_IFS after the packet sent.
2. Lower Tester sends a second DATA packet in the same Connection Event, using the minimum time between packets (T_IFS – 1.5 µsec) and setting MD bit to 0. Receive an empty DATA packet from the IUT T_IFS after the packet sent.
3. Repeat up to 10 Connection Events.
4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from preamble).

- **Expected Outcome**

  **Pass Verdict**

  The test procedure executes successfully, with the IUT maintaining the connection.

  The IUT responds at least to 95 percent of the packets sent by the Lower Tester in step 2.

- **Notes**

  Lower Tester is configured to use 1.5µsec instead of 2µsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.2.7 **LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]**

- **Test Purpose**

  Test that a slave IUT is able to maintain a connection with a master using the maximum timing between packets (T_IFS + 2 µsec).

  The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the maximum time required between packets.
• Reference

[3] 4.2

• Initial Condition
State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, any SCA value)

• Test Procedure
Execute the test using the common data channel selection parameters, a connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

![Diagram of connection process](image)

Figure 4.337: LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]

1. Lower Tester sends a DATA packet to the IUT at the beginning of the Connection Event, setting MD bit to 1. Receive an empty DATA packet from the IUT T_IFS after the packet sent.
2. Lower Tester sends a second DATA packet in the same Connection Event, using the maximum time between packets (T_IFS + 1.5 µsec) and setting MD bit to 0. Receive an empty DATA packet from the IUT T_IFS after the packet sent.
3. Repeat up to 10 Connection Events.
4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from preamble).

• Expected Outcome

Pass Verdict
The test procedure executes successfully, with the IUT maintaining the connection.
The IUT responds at least to 95 percent of the packets sent by the Lower Tester in step 2.
• Notes

Lower Tester is configured to use 1.5μsec instead of 2μsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.2.8 LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]

• Test Purpose

Test that the IUT initiates the Sleep Clock Accuracy Update procedure to change its Sleep Clock Accuracy. Verify that the connection remains active after the Sleep Clock Accuracy Update.

• Reference

[13] 4.6.25, 5.1.14

• Initial Condition

State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA and slave SCA values not the worst accuracy supported)

• Test Procedure

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action parameter set to 0x01 and receives an HCI_Command_Complete event with zero status.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the slaveSCA value that is worse than the initial slaveSCA from the connection process to the Lower Tester.
3. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT. The accuracy is worse than that currently in use.
4. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.

Figure 4.338: LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]
5. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
6. Repeat for a number of events (10 events) to conclude that the connection is active.

- Expected Outcome
  - Pass Verdict
    - The IUT receives 90 percent of the LL Data Channel PDUs from the Lower Tester in step 6 to conclude the connection is active.

4.4.2.9 LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update]

- Test Purpose
  Test that the IUT responds to the Sleep Accuracy Update initiated by the Lower Tester. Verify that the connection remains active after the Sleep Clock Accuracy Update.

- Reference
  [13] 4.6.25, 5.1.14

- Initial Condition
  State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, masterSCA value is 0 to 20ppm)

- Test Procedure

```
+----------------+      +----------------+      +----------------+
| Lower Tester  | <---  | IUT             | <---  | Upper Tester   |
+----------------+      +----------------+      +----------------+
                     |     | Connected Slave |     |                     |
                     |     | LL CLOCK_ACCURACY_REQ | LL CLOCK_ACCURACY_RSP |
                     |     | (masterSCA: 251-500 ppm) | (slaveSCA) |
                     |     | LL Data Channel |     | LL Data Channel |
                     |     | LL Data Channel |     | LL Data Channel |
```

Figure 4.339: LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update]

1. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
2. The IUT receives the LL_CLOCK_ACCURACY_REQ PDU to update the Sleep Clock Accuracy. The IUT switches to the updated master Sleep Clock Accuracy value and sends the LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester.
3. After the Lower Tester receives the LL_CLOCK_ACCURACY_RSP PDU the Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on
the same data channel. The Lower Tester sends the packets within 2 µs of the edge of the timing window.

4. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

5. Repeat for a number of events (100 events) to conclude that the connection is active. For each event the Lower Tester randomly selects the leading edge or the trailing edge in step 3.

• Expected Outcome
  
  Pass Verdict
  
  - The IUT receives at least 95 percent of the LL Data Channel PDUs from the Lower Tester in step 5 to conclude the connection is active.

4.4.2.10 LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy]

• Test Purpose
  
  Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester without reducing its Sleep Clock Accuracy.

• Reference
  
  [13] 4.6.25, 5.1.14

• Initial Condition
  
  State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, masterSCA value is 0 to 20ppm)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x00 and receives an HCI_Command_Complete event with status = 0x00 or 0x43.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT.
3. The Lower Tester responds with an LL_CLOCK_ACCURACY_RSP PDU.
4. The Upper Tester sends a command that will cause the IUT to initiate an LL procedure with the Lower Tester. The LL procedure must not be the Sleep Clock Accuracy procedure and must be one that requires a response.
5. The IUT sends the initial PDU of that procedure to the Lower Tester.
6. The Upper Tester sends HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x01 and receives an HCI_Command_Complete event with zero status.
7. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
8. The IUT sends the LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester. If it has not done so within 20 connection intervals, the Lower Tester completes the procedure started in step 4 and then continues to wait for the LL_CLOCK_ACCURACY_RSP PDU.

Figure 4.340: LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy]
• Expected Outcome
  
  Pass Verdict
  
  - The LL_CLOCK_ACCURACY_RSP PDU sent by the IUT in step 8 has the same or better sleep clock accuracy than that in the LL_CLOCK_ACCURACY_REQ in step 2.

4.4.3 MAS
Tests that the IUT behaves according to timing constraints in the master role. The master timing requirements apply only in active mode.

4.4.3.1 LL/TIM/MAS/BV-01-C [Earliest Transmissions to Master]

• Test Purpose
  
  Test that a master IUT is able to maintain a connection with a slave using the minimum inter-frame space in responses.

  The Lower Tester acts in the slave role, maintaining the connection and responding to the master using the minimum time required to respond to packets from the master. The Lower Tester applies the active mode timing requirements by using the maximum negative jitter for the slave packet response transmissions.

• Reference
  
  [3] 4.2

• Initial Condition
  
  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, common timeout, common channel map, not encrypted)

• Test Procedure
  
  Execute the test procedure using the common connection interval, slave latency and connection supervision timer.

![Connection Established. IUT Master Diagram](image)

**Figure 4.341:** LL/TIM/MAS/BV-01-C [Earliest Transmission to Master]
1. Lower Tester receives an empty DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.

2. In events where a response is required, Lower Tester sends a DATA packet in events required by the slave latency parameter only, using T_IFS=150-1.5 μs (maximum jitter value) and packet timing drift = -50 ppm (maximum positive clock drift) after the packet from the IUT.

3. Repeat steps 1–2 15 times.

4. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

- **Expected Outcome**

  **Pass Verdict**

  The test procedure executes successfully, with the IUT maintaining the connection.

- **Notes**

  Lower Tester is configured to use 1.5μsec instead of 2μsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.3.2 LL/TIM/MAS/BV-02-C [Latest Transmissions to Master]

- **Test Purpose**

  Test that a master IUT is able to maintain a connection with a slave using the maximum inter-frame space in responses.

  The Lower Tester acts in the slave role, maintaining the connection and responding to the master using the maximum time required to respond to packets from the master. The Lower Tester applies the active mode timing requirements by using the peak jitter for the slave packet response transmissions.

- **Reference**

  [3] 4.2

- **Initial Condition**

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, common timeout, common channel map, not encrypted)

- **Test Procedure**

  Execute the test procedure using the common connection interval, slave latency and connection supervision timer.
1. Lower Tester receives an empty DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.
2. In events where a response is required, Lower Tester sends a DATA packet in events required by the slave latency parameter only, using $T_{IFS}=150+1.5\ \mu s$ (minimum jitter value) and packet timing drift = $+50$ ppm (maximum negative clock drift) after the packet from the IUT.
3. Repeat steps 1–2 15 times.
4. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).

• Expected Outcome

**Pass Verdict**
The test procedure executes successfully, IUT maintaining the connection.

• Notes

Lower Tester is configured to use $1.5\mu sec$ instead of $2\mu sec$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

**4.4.3.3 LL/TIM/MAS/BV-03-C [Initiate Sleep Clock Accuracy Update]**

• Test Purpose
Test that the IUT initiates the Sleep Clock Accuracy Update procedure to change its sleep clock accuracy. Verify the accuracy of the timings of the connection event.

• Reference
[13] 4.6.25, 5.1.14

• Initial Condition
State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value not the best accuracy supported)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action parameter set to 0x00 and receives an HCI_Command_Complete event with zero status.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the masterSCA value that is better than the initial slaveSCA from the connection process to the Lower Tester.
3. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT.
4. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
5. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
6. Repeat for a number of events (10 events) to conclude the timing accuracy.

• Expected Outcome

Pass Verdict
- Verify that the connection event time intervals are within the timing range after the Sleep Clock Accuracy update.

4.4.3.4 LL/TIM/MAS/BV-04-C [Response to Sleep Clock Accuracy Update]

• Test Purpose

Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester. Verify the accuracy of the timings of the connection events.

• Reference

[13] 4.6.25, 5.1.14
• Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any master SCA value)

• Test Procedure

1. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT.
2. The IUT sends a LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester.
3. After the Lower Tester receives the LL_CLOCK_ACCURACY_RSP PDU the Lower Tester sends a correctly formatted LL Data Channel PDU, using the acknowledgement scheme, to the IUT on the same data channel.
4. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
5. Repeat for a number of events (10 events).

• Expected Outcome

Pass Verdict
- Verify that the connection event time intervals are within the timing range after the Sleep Clock Accuracy update.
- Verify that the Lower Tester receives the LL_CLOCK_ACCURACY_RSP with the master Sleep Clock Accuracy value.
- The LL_CLOCK_ACCURACY_RSP PDU sent by the IUT has the same or better sleep clock accuracy than the IUT was using at the start of the test.

4.4.3.5 LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy]

• Test Purpose
Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester without reducing its Sleep Clock Accuracy.

• Reference
[13] 4.6.25, 5.1.14
• **Initial Condition**
  
  State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value is the best supported by the IUT)

• **Test Procedure**

  ![Diagram of test procedure](image)

  **Figure 4.345: LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy]**

  1. The Upper Tester sends a command that will cause the IUT to initiate an LL procedure with the Lower Tester. The LL procedure must not be the Sleep Clock Accuracy procedure and must be one that requires a response.
  2. The IUT sends the initial PDU of that procedure to the Lower Tester.
  3. The Upper Tester sends `HCI_LE_Modify_Sleep_Clock_Accuracy` to the IUT with the Action parameter set to 0x01 and receives an `HCI_Command_Complete_event` with zero status.
  4. The Lower Tester sends an `LL_CLOCK_ACCURACY_REQ` PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
  5. The IUT sends the `LL_CLOCK_ACCURACY_RSP` PDU to the Lower Tester. If it has not done so within 20 connection intervals, the Lower Tester completes the procedure started in step 1 and then continues to wait for the `LL_CLOCK_ACCURACY_RSP` PDU.

• **Expected Outcome**

  **Pass Verdict**
  
  - The `LL_CLOCK_ACCURACY_RSP` sent by the IUT has the same or better sleep clock accuracy than the IUT was using at the start of the test.
4.4.3.6 LL/TIM/MAS/BV-06-C [Change Accuracy after Update]

- Test Purpose
  Test that when the sleep clock accuracy gets worse the accuracy updates after the Lower Tester responds with the LL_CLOCK_ACCURACY_RSP PDU.

- Reference
  [13] 4.6.25, 5.1.14

- Initial Condition
  State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value is the best supported by the IUT)

- Test Procedure
  Execute the test procedure using a connection interval of less than 500 ms.

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x01 and receives an HCI_Command_Complete event with a zero status.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the masterSCA value set to a worse accuracy.
3. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
4. Repeat for a number of events (50 events) to conclude that the timing accuracy has not changed.
5. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT.
6. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
7. Repeat for a number of events (10 events).

• Expected Outcome
  
  **Pass Verdict**
  - Verify that the timing of packets in steps 2–5 is no worse than the accuracy set in the initial condition.
  - Verify that the timing of packets in steps 6 and 7 are no worse than the accuracy specified in the LL_CLOCK_ACCURACY_REQ PDU.
  - The IUT receives at least 90 percent of the LL Data Channel PDUs from the Lower Tester in step 7 to conclude the connection is active.

4.4.3.7 LL/TIM/MAS/BV-07-C [Change Accuracy before Update]

• Test Purpose
  
  Test that when the sleep clock accuracy gets better the sleep clock accuracy is updated before the Lower Tester sends the LL_CLOCK_ACCURACY_RSP PDU.

• Reference
  
  [13] 4.6.25, 5.1.14

• Initial Condition
  
  State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value the worst supported by the IUT)

  Note: The SCA read in CONNECT_IND is not necessarily the worst supported by the IUT.

• Test Procedure
  
  Execute the test procedure using a connection interval of less than 500 ms.
1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x00 and receives an HCI_Command_Complete event with zero status.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the masterSCA value set to a better accuracy.
3. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
4. Repeat for a number of events (50 events) to conclude that the timing accuracy has changed.
5. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT.
6. The Lower Tester receives correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
7. Repeat for a number of events (10 events).

- Expected Outcome

Pass Verdict
- Verify that the timing of packets in steps 3–7 is no worse than the accuracy specified in step 2.
- The IUT receives at least 90 percent of the LL Data Channel PDUs from the Lower Tester in step 7 to conclude the connection is active.
4.4.4 SCN
Tests that the IUT behaves according to timing constraints as a scanner.

4.4.4.1 Extended Scanning, Secondary Channel, Earliest Transmission to Scanner

• Test Purpose
Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the minimum allowed timing (T_MAFS) on the secondary channel.

• Reference
[10] 4.4.2.5.4

• Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
• Test Procedure

1. Configure Lower Tester to start scannable advertising using PHY under test on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
2. Upper Tester enables active scanning IUT with the Scanning PHYs field set as specified in Table 4.95.
3. Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the minimum time after the end of the ADV_EXT_IND PDU. The Aux Offset in the AuxPtr shall be the smallest representable value that is at least the length of the packet plus T_MAFS and the AUX_ADV_IND shall be transmitted at the represented time, which may be between 300 and 329 µs after the end of the packet. For each extended advertising event the Lower Tester shall change the DID value of ADI field.
4. The Lower Tester receives an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the minimum time after the end of the AUX_SCAN_REQ PDU (T_IFS – 1.5 µsec).
5. The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.

Figure 4.348: Extended Scanning, Secondary Channel, Earliest Transmission to Scanner
6. Repeat steps 3–5 until the Lower Tester has received 100 AUX_SCAN_REQ PDUs from the IUT.
7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

• Test Case Configuration

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHYs</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.4.4.1.1</strong> LL/TIM/SCN/BV-01-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner – LE 1M PHY, Core 5.0]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.4.4.1.2</strong> LL/TIM/SCN/BV-04-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner – LE 2M PHY, Core 5.0]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (LE 2M PHY)</td>
</tr>
<tr>
<td><strong>4.4.4.1.3</strong> LL/TIM/SCN/BV-05-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner – LE 1M PHY, Core 5.1]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x01 (LE 1M PHY)</td>
</tr>
<tr>
<td><strong>4.4.4.1.4</strong> LL/TIM/SCN/BV-06-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner – LE 2M PHY, Core 5.1]</td>
<td>0x01 (LE 1M PHY)</td>
<td>0x02 (LE 2M PHY)</td>
</tr>
</tbody>
</table>

Table 4.95: Extended Scanning, Secondary Channel, Earliest Transmission to Scanner Test Cases

• Expected Outcome

Pass Verdict
- For each AUX_SCAN_REQ PDU received and responded to by the Lower Tester, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester, the Advertising_SID value in the event is the value as specified in Table 4.95.
4.4.4.2  Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY

- Test Purpose
Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the minimum allowed timing (T_MAFS) and AUX_SCAN_RSP PDUs when the PDU is sent at the minimum allowed timing (T_IFS - 2 µs) using the LE Coded PHY on the secondary channel.

- Reference
[10] 4.4.2.5.4

- Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
1. Configure Lower Tester to start scannable advertising using the LE Coded PHY on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
2. Upper Tester enables active scanning IUT with the Scanning PHYs field set to 0x04 (LE Coded PHY).
3. Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the minimum time after the end of the ADV_EXT_IND PDU. The Aux Offset in the AuxPtr shall be the smallest representable value that is at least the length of the packet plus T_MAFS and the AUX_ADV_IND shall be transmitted at the represented time, which may be between 300 and 329 µs after the end of the packet. For each extended advertising event the Lower Tester shall change the DID value of ADI field.
4. The Lower Tester receives an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the minimum time after the end of the AUX_SCAN_REQ PDU (T_IFS - 1.5 µs).
5. The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.

Figure 4.349: Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY
6. Repeat steps 3–5 until the Lower Tester has received 100 AUX_SCAN_REQ PDUs from the IUT.
7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

- Expected Outcome
  **Pass Verdict**
  - For each AUX_SCAN_REQ PDU received and responded to by the Lower Tester, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
  - In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value as specified in Table 4.96.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/TIM/SCN/BV-02-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY, Core 5.0]</td>
<td>0xFF or the Advertising SID from the AUX_ADV_IND PDU.</td>
</tr>
<tr>
<td>LL/TIM/SCN/BV-07-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY, Core 5.1]</td>
<td>Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU.</td>
</tr>
</tbody>
</table>

Table 4.96: Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY test cases

- Notes
  Lower Tester is configured to use 1.5 µs instead of 2 µs. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.4.3 Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY

- Test Purpose
  Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the maximum allowed timing (Aux Offset + 1 Offset Unit) and AUX_SCAN_RSP PDUs when the PDU is sent at the maximum allowed timing (T_IFS + 2 µs + simulated range delay) using the LE Coded PHY on the secondary channel.

- Reference
  [10] 4.2.3, 4.4.2.5.4

- Initial Condition
  Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map
  State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
Test Procedure

1. Configure Lower Tester to start scannable advertising using the LE Coded PHY on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
2. Upper Tester enables active scanning IUT with the Scanning PHYs field set to 0x04 (LE Coded PHY).
3. Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the maximum time after the end of the ADV_EXT_IND PDU (Aux Offset + Offset Unit - 0.5 µs). For each extended advertising event the Lower Tester shall change the DID value of ADI field.
4. The Lower Tester receives an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the maximum time after the end of the AUX_SCAN_REQ PDU (T_IFS + 1.5 µs + simulated range delay), where simulated range delay is calculated as $2 \times TSPX_{\text{scan\_max\_coded\_range}} \times 4 \text{ ns}$ with the result floored to the nearest multiple of 0.5 µs that is less than or equal to the result.
5. The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.

Figure 4.350: Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY
6. Repeat steps 3–5 until the Lower Tester has received 100 AUX_SCAN_REQ PDUs from the IUT.
7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.

• Expected Outcome

Pass Verdict
- For each AUX_SCAN_REQ PDU received and responded to by the Lower Tester, the Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester in step 5, the Advertising_SID value in the event is the value as specified in Table 4.97.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Advertising SID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.4.4.3.1</strong> LL/TIM/SCN/BV-03-C [Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY, Core 5.0]</td>
<td>0xFF or the Advertising SID from the AUX_ADV_IND PDU.</td>
</tr>
<tr>
<td><strong>4.4.4.3.2</strong> LL/TIM/SCN/BV-08-C [Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY, Core 5.1]</td>
<td>Advertising SID from the AUX_SCAN_RSP PDU or, if absent, the AUX_ADV_IND PDU.</td>
</tr>
</tbody>
</table>

Table 4.97: Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY test cases

• Notes

Lower Tester is configured to use 1.5 µs instead of 2 µs and Offset Unit - 0.5 µs instead of Offset Unit. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.5 **ENC**

Tests that the IUT rejects packets with an invalid preamble or access address.

There is a possibility that the error injected in the access address is reverted on the radio path, for this the tests transmit the packet with the invalid access address more than once and typically reject behavior where the IUT responds over a particular count. Note that the pass criteria rely on a particular bit error rate for verdict assignment.

The access addresses used in the tests, while invalid data for the scenarios, are still formulated according to the access address formulation rules.

4.5.1 **ADV**

Tests that the IUT rejects packets with an invalid preamble or access address as an advertiser.

4.5.1.1 **LL/ENC/ADV/BV-01-C [Scan Request Invalid Address]**

• Test Purpose

Test that an advertiser IUT ignores a scan request with an invalid advertising channel synchronization word and continues advertising.
The Lower Tester sends the invalid scan request and observes the IUT continuing advertising.

- **Reference**
  
  [3] 2.1.2, 4.4.2, 3.1

- **Initial Condition**

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser)

- **Test Procedure**

  Execute the test procedure with an advertising interval between the minimum and maximum advertising intervals.

---

**Figure 4.351: LL/ENC/ADV/BI-01-C [Scan Request Invalid Address]**
1. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising intervals and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)’ if filtering policy is supported.
2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT.
3. Lower Tester sends a SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a White Listed device address as parameter with an invalid access address from corrupting a single bit in the access address after receiving an ADV_IND packet from IUT on the advertising channel configured in step 2. The SCAN_REQ packet is sent T_IFS after the end of an ADV_IND packet.
4. Lower Tester expects the IUT to continue advertising with no response to the SCAN_REQ packet.
5. Repeat steps 3–4 40 times, each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 SCAN_REQ packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 in that order; for the other 20 SCAN_REQ packets, it shall corrupt bits 31 to 12 in that order.
6. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to the IUT to disable advertising and receives an HCI_Command_Complete event in response.

• Test Condition
The parameters in this test are calculated for a BER of 0.1 percent or better.

• Expected Outcome
Pass Verdict
The test procedure completes with the IUT responding to not more than 8 of the 40 SCAN_REQ packets sent by the Lower Tester in step 3.

• Notes
This test verifies the reverse of a positive requirement.
The count of responses (more than 8 out of 40) representing the single bit error reverting and the rest of the packet intact is calculated not to occur in practice using a binomial distribution.

Bit 0 is the first bit transmitted over the air.

4.5.1.2 LL/ENC/ADV/BI-02-C [Master Packets Invalid Address]

• Test Purpose
Test that an advertiser IUT receives a connection request, stops advertising after reception and after transmissions with an access address different from the connection request from the master until the connection supervision timer expires, considers the connection setup failed.
The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role but with packets with a mismatching access address.

• Reference
[3] 2.1.2, 4.4.4, 3.1

• Initial Condition
Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.
State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser)

• Test Procedure

Execute the test procedure using the common data channel selection parameters, zero slave latency and a connection interval one tenth of the connection supervision timeout value (10 ms and 100 ms) in the connection request.

1. Upper Tester enables undirected advertising in the IUT using one selected advertising channel (defined as an IXIT) and filtering policy set to ‘Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)’ if filtering policy is supported.
2. Configure Lower Tester to initiate a connection.
3. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
4. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.
5. Lower Tester sends an empty DATA packet with an access address different from the connection request (corrupting a single bit), starting the first event using the common connection interval timing after the connection request.
6. Lower Tester receives no response from the IUT.
7. Repeat steps 5–6 until a time period equal to twice the connection supervision timeout value or until step 8 executes:

8. Upper Tester receives an HCI_Disconnection_Complete event indicating that creation of the connection failed (connection has failed to be established), with the connection handle parameter matching to step 4.

9. Repeat steps 1–8 40 times, each time changing the position of the bit which is corrupted in the access address as follows: In the first 20 iterations, the Lower Tester shall corrupt bits 1 to 20 in that order; for the other 20 iterations, it shall corrupt bits 31 to 12 in that order.

• Test Condition
The parameters in this test are calculated for a BER of 0.1 percent or better.

• Expected Outcome
Pass Verdict
For each of the 40 connection attempts, the test procedure completes with the IUT stopping advertising, moving to the data channel and reporting the failure to maintain the connection. The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete_Event with reason code 0x3E is received in at least 32 of 40 connection attempts.

• Notes
The advertising channel used for the connection setup is assumed not to influence the access address recognition in the data channels.

The probability calculations describing the range of responses to expect are described in the test LL/ENC/SLA/BI-01-C [Packets to another Slave].

4.5.2 SCN
Tests that the IUT rejects packets with an invalid preamble or Access Address as a scanner.

4.5.2.1 LL/ENC/SCN/BI-01-C [Passive Scanning Invalid Address]

• Test Purpose
Test that a passive scanner IUT ignores advertising indication packets with invalid channel addresses.

The Lower Tester sends advertising packets with an invalid advertising channel synchronization word and checks that the IUT does not react to them.

• Reference
[3] 2.1.2, 4.4.3.1, 3.1

• Initial Condition
State: Passive Scanning (selected scan interval, selected scan window AND (All White Listed (policy for scanner)

• Test Procedure
Execute the test procedure with a timing combination using the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising
interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).

1. Upper Tester enables passive scanning in the IUT.
2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), and a white listed device address.
3. Lower Tester sends an ADV_NONCONN_IND packet with an invalid access address, each advertising event using the selected advertising channel only, timing the events to match the scan interval.
4. Upper Tester receives no HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester.
5. Repeat steps 3–4 for a time that exceeds a number of scan intervals (40), each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 ADV_NONCONN_IND packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 of the access address in that order. For the other 20 ADV_NONCONN_IND packets, it shall corrupt bits 31 to 12 of the access address in that order.
6. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and receives an HCI_Command_Complete event in response.

- Test Condition
  The parameters in this test are calculated for a BER of 0.1 percent or better.
- Expected Outcome
  **Pass Verdict**
  The test procedure completes with the IUT reporting not more than 8 of the 40 advertising packets sent in step 3.
• Notes

Bit 0 is the first bit transmitted over the air.

The possibility for a scanner device to respond to an invalid radio frame (probability calculations described in the test LL/ENC/SLA/BI-01-C [Packets to another Slave] provide for the case that an error injected into the access address is reverted. While not likely, such a bit error may occur in test execution, which would make a test not accepting any scan reports unreliable. Because the probability for the scanner to be listening the frequency of the advertising packet is unknown, this is ignored in the calculations. This test fails implementations not checking all the bits in the access address.

4.5.2.2 LL/ENC/SCN/BI-02-C [Active Scanning Invalid Address]

• Test Purpose

Test that an active scanner IUT ignores advertising indication packets with invalid channel addresses. The Lower Tester sends advertising packets with an invalid access address and checks that the IUT does not react to them.

• Reference

[3] 2.1.2, 4.4.3.2, 3.1

• Initial Condition


State: Active Scanning (public address, selected scan interval, selected scan window) AND (All White Listed (policy for scanner))

• Test Procedure

Execute the test procedure using a scan interval 3 times the length of the average advertising interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).
1. Upper Tester enables active scanning in the IUT.
2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), undirected advertising, and a white listed device address.
3. Lower Tester sends an ADV_IND packet with an invalid access address, each advertising event using the selected advertising channel only, using the selected advertising interval.
4. Lower Tester receives no SCAN_REQ packet after T_IFS in response to any of the ADV_IND packets.
5. Upper Tester receives no HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester.
6. Repeat steps 3–5 until the time exceeds a number of scan intervals (40), each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 ADV_IND packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 of the access address in that order. For the other 20 ADV_IND packets, it shall corrupt bits 31 to 12 of the access address in that order.
7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete_event in response.

**Test Condition**

The parameters in this test are calculated for a BER of 0.1 percent or better.

**Expected Outcome**

**Pass Verdict**

The test procedure completes with the IUT reporting not more than 8 of the 40 advertising indications sent in step 3.
• Notes

Bit 0 is the first bit transmitted over the air.

4.5.3 INI

Test that an initiator IUT sends a connection request to an advertiser and receiving reply transmissions with an access address different from the connection request for 6 connection intervals, considers the connection setup failed.

Note that there is no initiator test for advertising packets as the device class dependencies require that the scanner role is also implemented in a Controller with the initiator role.

4.5.3.1 LL/ENC/INI/BI-01-C [Slave Packets Invalid Address]

• Test Purpose

Test that an initiator IUT sends a connection request to an advertiser and receives reply transmissions with an access address different from the connection request for six (6) connection intervals, and considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, then begins to maintain the connection but with packets using a mismatching address.

• Reference

[3] 2.1.2, 4.5, 3.1

• Initial Condition


State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

• Test Procedure

Execute the test procedure using a selected advertising channel, advertising on the selected advertising channel only, using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval, latency and timeout to setup the connection.
1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiating in the IUT.
3. Lower Tester sends ADV_IND packets and receives a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
4. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
5. After the CONNECT_IND packet, receive the first empty DATA packet from the IUT on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
6. Lower Tester sends an empty DATA packet with an access address different from the one in the connection request (corrupting a single bit in the access address), using the acknowledgement scheme, to the IUT on the same data channel.
7. Lower Tester receives following empty DATA packets on subsequent data channels.
8. Repeat steps 6–7 until the IUT stops.
9. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating that a connection has failed to be established and with the connection handle matching to the one used in the connection.
10. Repeat steps 1–9 40 times, with each iteration through the loop changing the bit position of the corrupted bit in the access address as follows. For the first 20 iterations, the Lower Tester shall...
corrupt bits 1 to 20 in that order. For the other 20 iterations, the Lower Tester shall corrupt bits 31 to 12 in that order.

- **Test Condition**
  The parameters in this test are calculated for a BER of 0.1 percent or better.

- **Expected Outcome**
  **Pass Verdict**
  The test procedure completes the iterations using each access address with the IUT sending a connection request, then continuing master transmissions for six (6) connection intervals.

  The IUT reports the failure to setup the connection with an HCI_Disconnection_Complete_Event with reason code 0x3E.

  The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete event with reason code 0x3E is received in at least 32 of 40 connection attempts.

- **Notes**
  Bit 0 is the first bit transmitted over the air.

### 4.5.4 SLA
Tests that the IUT rejects packets with an invalid preamble or access address as a slave.

#### 4.5.4.1 LL/ENC/SLA/BI-01-C [Packets to another Slave]

- **Test Purpose**
  Tests that a slave IUT ignores a packet starting an event belonging to a different connection.

  The Lower Tester acts in the master role in the connection, first setting up the connection with valid packets then changing the access address and observes the slave packets on the data channels in use.

- **Reference**
  [3] 2.1.2, 4.5, 3.1

- **Initial Condition**
  Parameters: LL_slave_connSlaveLatency_MIN

  State: Connected Slave (common connection interval, 0 slave latency, common timeout, common channel map, any SCA value)
• Test Procedure

1. The Lower Tester prepares a DATA packet, corrupting the second bit in the access address.
2. The Lower Tester sends the DATA packet to the IUT using the data channel selection parameters.
3. The Lower Tester receives no DATA packet T_IFS after the packet sent.
4. The Lower Tester prepares a DATA packet, corrupting the next bit in sequence (increasing bit number) in the access address.
5. Repeat steps 2–4 until a total of 20 DATA packets have been sent in consecutive connection intervals.
6. The Lower Tester stops sending DATA packets and allows a connection supervision timeout.
7. The Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.
8. A connection is established (IUT acts as slave).
9. The Lower Tester prepares a DATA packet, corrupting the last bit in the access address.
10. The Lower Tester sends the DATA packet to the IUT using the data channel selection parameters.
11. The Lower Tester receives no DATA packet T_IFS after the packet sent.
12. The Lower Tester prepares a DATA packet, corrupting the next bit in sequence (decreasing bit number) in the access address.
13. Repeat steps 10–12 until a total of 20 DATA packets have been sent in consecutive connection intervals.
14. The Lower Tester stops sending DATA packets and allows a connection supervision timeout.
15. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.

• Expected Outcome

Pass Verdict

For each sequence of 20 DATA packets, the IUT responds in no more than 4 of the 20 connection events.

• Notes

The test allows the single bit error reverting by observing a time period longer than the connection supervision timeout. With the probability of receiving the preamble and access address correctly around 96% and the relation of the connection interval to the connection supervision timeout value giving 10 attempts before failing, the test procedure does in practice not terminate to connection...
supervision timeout value under the bit error rate conditions required for an IUT not using the entire access address.

The probability of the error in the access address reverted is around 0.1 percent, so for 10 events received correctly and none incorrectly, the probability using binomial distribution is around 99 percent. Using geometric distribution to represent the number of attempts before an occurrence of 10 correctly received invalid access addresses in a row, the expected value for the repetition count is below 1 (around 0.01), so 20 events are sufficient to conclude the test procedure. This means that an IUT using the entire access address will terminate the connection.

4.5.5 MAS
Tests that the IUT rejects packets with an invalid preamble or access address procedures in the master role.

4.5.5.1 LL/ENC/MAS/BI-01-C [Packets to another Master]

- Test Purpose
  Tests that a master IUT ignores packets not belonging to the connection transmitted using a different access address.

  The Lower Tester acts as a slave maintaining a connection, then starts to respond with packets using an invalid access address.

- Reference
  [3] 2.1.2, 4.5, 3.1

- Initial Condition
  Parameters: LL_master_connSlaveLatency_MIN

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
• Test Procedure

1. A connection is established (IUT acts as master).
2. Lower Tester receives a DATA packet, once a connection interval from the IUT on the data channel derived from the selection parameters.
3. Lower Tester sends a DATA packet with an incorrect access address by corrupting a single bit in the access address.
4. Repeat steps 1–2 for a time period equal to 2 times the connection supervision timeout value from the last valid packet sent to which an acknowledgement was received.
5. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.
6. Repeat steps 1–5 40 times, with each iteration through the loop changing the bit position of the corrupted bit in the access address as follows. For the first 20 iterations, the Lower Tester shall corrupt bits 1 to 20 in that order. For the next 20 iterations, the Lower Tester shall corrupt bits 31 to 12 in that order.

• Expected Outcome

Pass Verdict
Each test procedure iteration executes with the parameters selected, with the connection terminated within the time period of 2 times the connection supervision timeout value from the last valid packet sent.

The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete_Event with reason code 0x08 is received in at least 32 of 40 connections.

• Notes

Bit 0 is the first bit transmitted over the air.
4.6 FRH
Tests that the IUT behaves according to the data channel selection requirements.

4.6.1 Common PDU Contents
The packet descriptions for Lower Tester sent and Lower Tester accepted data channel packets are below.

CHANNEL_MAP_IND CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>LSO MSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isb msb</td>
<td>Isb msb</td>
</tr>
<tr>
<td>ChM</td>
<td>Instant</td>
</tr>
<tr>
<td>(5 octets)</td>
<td>(2 octets)</td>
</tr>
</tbody>
</table>

4.6.2 ADV
Tests that the IUT behaves according to the data channel selection procedure during connection setup procedure in the advertiser role.

4.6.2.1 LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]

- Test Purpose
  Test that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role using the correct data channel sequence.

  The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role.

- Reference
  [3] 4.5.8

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name)
**Test Procedure**

1. Configure Lower Tester to initiate a connection.
2. Upper Tester enables advertising in the IUT using one advertising channel (defined as an IXIT).
3. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet, T_IFS after the end of the advertising packet, using a maximum hop length parameter.
4. Upper Tester receives an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.
5. Lower Tester sends an empty LL DATA packet every connection interval, using the data channel selection parameters.
6. Lower Tester receives an empty LL DATA packet from the IUT T_IFS after the data packet sent on the same data channel. Lower Tester continues sending empty data packets using the acknowledgement scheme.
7. Repeat step 5–6 until receiving a response per each channel in use, or until the channel has been in use 15 times with no response.
8. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 4).
9. Repeat steps 1–2.
10. Lower Tester receives an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet using a minimum hop length parameter, T_IFS after the end of the advertising packet.
11. Repeat steps 4–8.

Figure 4.358: LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]
• Expected Outcome
  Pass Verdict
  The test procedure executes successfully with the data channel selection input variations.
  The IUT maintains the connection.
  The IUT responds on all channels at least once.

4.6.3  SLA
Tests that the IUT behaves according to the data channel selection procedure in the slave role.

4.6.3.1 LL/FRH/SLA/BV-01-C [Accepting Channel Map Update]

• Test Purpose
  Test that a slave IUT accepts a channel map update request from the master and adopts the new
  channel map at the correct time is able to maintain the connection.
  The Lower Tester acts as a master. It maintains the connection and requests a new channel map to
  be taken into use. It then observes the IUT accepting and adopting the channel map.

• Reference
  [3] 4.5.8, 5.1.2

• Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, public address, any
  advertising channel map, common connection interval, common slave latency, common timeout,
  selected channel map, any SCA value)
• Test Procedure

1. Lower Tester sends a CHANNEL_MAP_IND packet to the IUT using every second channels used as channel map and using an event count of 100. Receive an acknowledgement from the IUT.

2. Lower Tester sends an empty DATA packet to the IUT and receives a response in all events until the time of the update. Repeat until the event count matches the indicated time of connection update.

3. At the time of the update start maintaining the connection with the new parameters.

4. Lower Tester sends an empty LL DATA packet every connection interval, using the data channel selection parameters. Receive an empty LL DATA packet from the IUT, T_IFS after the data
packet sent on the same data channel. Lower Tester continues to send empty data packets using the acknowledgement scheme. Repeat until receiving a response per each channel in use, or until the channel has been in use 15 times.

5. Lower Tester sends a CHANNEL_MAP_IND packet to the IUT using all channels used as channel map and using an event count of 100. Receive an acknowledgement from the IUT.

6. Repeat steps 2–4

7. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble step execution).

- Expected Outcome

**Pass Verdict**

The test procedure executes successfully, with the IUT acknowledging the channel map update requests and adopting the updated data channel selection parameters at the assigned event,

The IUT responds in 65 of the 100 events between the channel map request and the assigned event to adopt the updated data channel selection parameters.

- Notes

The channels applied in this test are not parameterized, assuming that the test inputs will select any channels not used.

4.6.3.2 **LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2]**

- Test Purpose

Tests that a slave IUT accepts a channel map update request from the master while using Channel Selection Algorithm #2, adopts the new channel map at the correct time, and maintains the connection.

The Lower Tester acts as master. It maintains the connection, requests a new channel map to be taken into use, and observes the IUT accepting and adopting the new channel map.

- Reference

[10] 4.5.8, 5.1.2

- Initial Condition

State: Connected Slave (Channel Selection Algorithm #2, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value)
• Test Procedure

**Figure 4.360: LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2]**

1. The Lower Tester sends a CHANNEL_MAP_IND packet to the IUT, with the ChM field set to use every second channel (every other channel bit set to ‘1’, others set to ‘0’) and the Instant field set to 100 connection events ahead.
2. The Lower Tester sends an empty LL DATA packet to the IUT every connection interval and receives a response using the acknowledgement scheme. Repeat until the event count matches the indicated time of connection update.
3. Upon connection update, the Lower Tester switches to selecting data channel indices using Channel Selection Algorithm #2 with the new channel map.
4. The Lower Tester sends an empty LL Data packet to the IUT every connection interval and receives a response using the acknowledgement scheme. Repeat until the Lower Tester has receives a response for each channel in use, or until a channel has been used 15 times.
5. Repeat steps 1–4, except in step 1, the ChM field shall be set to use all channels.

• Expected Outcome

**Pass Verdict**

The test procedure executes successfully, with the IUT accepting the channel map update request and maintaining the connection using Channel Selection Algorithm #2 with the new channel map.

The IUT sends and receives data each connection event using data channel indices selected by the Channel Selection Algorithm #2 using the new channel map.

**4.6.4 MAS**

Tests that the IUT behaves according to the data channel selection procedure in the master role.

**4.6.4.1 LL/FRH/MAS/BV-01-C [Requesting Channel Map Update]**

• Test Purpose

Test that a master IUT performs the channel map updating procedure.

The Lower Tester acts in the slave role, requests the IUT through the HCI to perform the updating procedure, accepts the request sent by the IUT and adopts the new channel map.
- **Reference**
  
  [3] 4.5.8, 5.1.2

- **Initial Condition**

  Parameters: LL_master_Channel_Map

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- **Test Procedure**

  ![Diagram](image)

  **Figure 4.361: LL/FRH/MAS/BV-01-C [Requesting Channel Map Update]**

  1. Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT, including ‘supported even channels’ data channel selection parameters. Receive an HCI_Command_Complete event from the IUT in response.

  2. Lower Tester receives an LL_CHANNEL_MAP_IND control packet from the IUT, with a channel map different than the one currently in use but not necessarily matching the parameters submitted in step 1, and sends an empty DATA packet to the IUT on the same data channel using the acknowledgement scheme. If the Lower Tester does not receive the LL_CHANNEL_MAP_IND, skip steps 3–5.

  3. Maintain the connection using empty DATA packets. Repeat until the event count matches the instant indicated in the channel map update request.

  4. Upon the instant, take the new parameters in use.

  5. Lower Tester receives an empty LL DATA packet from the IUT and sends an empty LL DATA packet every connection interval. Continue to send empty data packets using the acknowledgement scheme. Repeat until receiving a response per each channel in use. If any channel has been used 15 times without receiving a response on that channel, this test ends in a Fail Verdict.
6. Repeat steps 1–5, except that in step 1, the Upper Tester includes 'all channels used' data channel selection parameters.

• Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT requesting the channel map update, maintaining the connection using the original data channel selection parameters before the instant, and maintaining the connection with the updated data channel selection parameters after the instant.

Inconclusive Verdict

For both iterations, either the IUT does not send an LL_CHANNEL_MAP_IND PDU in step 2 or the map in the PDU is the same as the one currently in use.

4.6.4.2 LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2]

• Test Purpose

Tests that a master IUT performs the channel map update procedure while using Channel Selection Algorithm #2.

The Lower Tester acts in the slave role, requests the IUT through the HCI to perform the updating procedure, accepts the request sent by the IUT, and adopts the new channel map using Channel Selection Algorithm #2.

• Reference

[10] 4.5.8, 5.1.2

• Initial Condition

Parameters: LL_master_Channel_Map

State: Connected Master (Channel Selection Algorithm #2, any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).
• **Test Procedure**

![Diagram of test procedure](image)

**Figure 4.362: LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2]**

1. The Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT with Channel_Map set to mark only supported even channels as unknown (supported even channel bits set to ‘1’, others set to ‘0’) and receives an HCI_Command_Complete event in response.

2. The Lower Tester receives an LL_CHANNEL_MAP_IND packet from the IUT, with a channel map different than the one currently in use but not necessarily matching the parameters submitted in step 1, and sends an empty LL DATA packet to the IUT on the same channel using the acknowledgement scheme. If the Lower Tester does not receive the LL_CHANNEL_MAP_IND, skip steps 3–5.

3. The Lower Tester maintains the connection using empty LL DATA packets. Repeat until the event count matches the instant indicated in the channel map update request.

4. Upon the instant, the Lower Tester switches to selecting data channel indices using Channel Selection Algorithm #2 with the new channel map.

5. The Lower Tester receives LL DATA packets from the IUT and sends empty LL DATA packets every connection interval using the acknowledgement scheme. Repeat until the Lower Tester has received a response for each channel in use. If any channel has been used 15 times without receiving a response on that channel, this test ends in a Fail Verdict.

6. Repeat steps 1–5, except in step 1, the Channel_Map shall be set to mark all supported channels as unknown (supported channel bits set to ‘1’, others set to ‘0’).

• **Expected Outcome**

**Pass Verdict**

The test procedure executes successfully, with the IUT requesting the channel map update, maintaining the connection using Channel Selection Algorithm #2 with the original data channel selection parameters before the instant, and maintaining the connection using Channel Selection Algorithm #2 with the updated data channel selection parameters after the instant.

The IUT sends and receives data each connection event using data channel indices selected by the Channel Selection Algorithm #2 using the new channel map.
Inconclusive Verdict
For both iterations, either the IUT does not send an LL_CHANNEL_MAP_IND PDU in step 2 or the map in the PDU is the same as the one currently in use.

4.6.4.3 LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels]

- Test Purpose
Tests that a master IUT performs the Minimum Number of Used Channels Procedure. The Lower Tester acts in the slave role, sending the request to the IUT and maintaining the connection.

- Reference
[10] 5.1.11

- Initial Condition
Parameters: LL_master_Channel_Map
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

![Diagram](Figure 4.363: LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels])

1. The Lower Tester sends an LL_MIN_USED_CHANNELS_IND PDU to the IUT. For each PHY the IUT supports, the matching bit in the PHYS field shall be set to ‘1’. The MinUsCh field shall be set to 0x25 (37 channels).
2. The Lower Tester receives an LL Data packet from the IUT every connection interval and sends a response using the acknowledgement scheme. Repeat 10 times.

- Expected Outcome
Pass Verdict
The test procedure executes successfully, with the IUT receiving the LL_MIN_USED_CHANNELS_IND PDU and maintaining the connection.
4.7 PAC
Tests that the IUT rejects packets with invalid encoding contents.

4.7.1 Common PDU Contents
UNKNOWN_RSP CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
</tr>
</tbody>
</table>

UnknownType
(1 octet)

4.7.2 SLA
Tests that the IUT rejects packets with invalid encoding contents in the slave role.

4.7.2.1 LL/PAC/SLA/BV-01-C [Unknown Packet from Master]

- Test Purpose
  Test that a slave IUT responds with the unknown response packet to a device transmitting a control packet not in the supported specification or not supported by the IUT.

  The Lower Tester acts in the master role and starts an event with packet including a control type indicator not used in the supported specification, and then observes the IUT response. The Lower Tester repeats this test for all opcodes that are not supported by the IUT.

- Reference
  [3] 2.4.2, 2.4.2.8

- Initial Condition
  Parameters: <The feature set supported by the IUT, if the specification features are not selectable for this test.>

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)
• **Test Procedure**

![Diagram](image)

**Figure 4.364: LL/PAC/SLA/BV-01-C [Unknown Packet from Master]**

1. Lower Tester sends a control packet with an invalid control type value or a control type value not supported by the IUT to the IUT. This includes having the Lower Tester send opcodes that are invalid for the current role (e.g., LL_SLAVE_FEATURE_REQ sent to a slave IUT).
2. Lower Tester receives an LL_UNKNOWN_RSP, LL_REJECT_EXT_IND, or LL_REJECT_IND PDU packet from the IUT with an error code different than success.
3. Repeat steps 1–2 for all invalid control type values or control type values not supported by the IUT.
4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble step execution).

• **Expected Outcome**

**Pass Verdict**

The IUT sends an unknown response packet upon reception of all invalid/unsupported control packets.

**4.7.3 MAS**

Tests that the IUT rejects packets with invalid encoding contents in the master role.

**4.7.3.1 LL/PAC/MAS/BV-01-C [Unknown Packet from Slave]**

• **Test Purpose**

Test whether a master IUT responds with the unknown response packet to a device transmitting a control packet not in the supported specification or not supported by the IUT.

The Lower Tester acts in the slave role, maintaining a connection, responding to the master with a packet including a control type indicator not used in the supported specification, then observes the IUT response. The Lower Tester repeats this test for all opcodes that are not supported by the IUT.

• **Reference**

[3] 2.4.2, 2.4.2.8
• **Initial Condition**

Parameters: "The feature set supported by the IUT, if the specification features are not selectable for this test."

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

• **Test Procedure**

![Diagram](image)

1. Lower Tester receives a DATA packet from the IUT and responds sending a control packet with an invalid control type value or a control type value not supported by the IUT to the IUT. This includes having the Lower Tester send opcodes that are invalid for the current role (e.g., LL_CONNECTION_UPDATE_IND or LL_CHANNEL_MAP_IND sent to a master IUT).
2. Lower Tester receives an LL_UNKNOWN_RSP, LL_REJECT_EXT_IND, or LL_REJECT_IND PDU packet from the IUT in the next event with an error code different than success.
3. Repeat steps 1–2 for all invalid control type values or control type values not supported by the IUT.
4. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble steps).

• **Expected Outcome**

**Pass Verdict**

The IUT sends an unknown response packet upon reception of all invalid/unsupported control packets.

### 4.7.4 Both Connected Roles

**4.7.4.1 [Control PDUs with Invalid Length]**

Test that the IUT correctly handles invalid LL Control PDUs.

The Lower Tester sends invalid LL Control PDUs to the IUT and checks that it replies with a valid response.
• Test Case IDs
  LL/PAC/SLA/BI-01-C
  LL/PAC/MAS/BI-01-C

• Reference
  [3] Sections 2.4.2, 2.4.2.8

• Initial Condition
  State: Connected in the relevant role (any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, common timeout, common channel map, any SCA value)

• Test Procedure

1. Execute steps 2–9 for each LL Control PDU opcode supported by the IUT.
2. If the valid length of CtrData for the current opcode is greater than 0, execute steps 5–9 for CtrData length equal to 0 and for CtrData length equal to (the valid length - 1).
3. If the valid length of CtrData for the current opcode is less than 26, execute steps 5–9 for CtrData length equal to (the valid length + 1) and for CtrData length equal to 26.
4. Execute steps 5–9 for additional random and unique invalid CtrData lengths until 6 different invalid lengths have been used for the current opcode.
5. The Lower Tester sends an LL Control PDU to the IUT with the current opcode and length of CtrData, using random octets for the CtrData.
6. The Lower Tester expects one of the following to happen, as specified in Table 4.98:
   a. The Lower Tester receives an LL_UNKNOWN_RSP PDU from the IUT specifying the opcode of the PDU sent in step 5.
   b. The Lower Tester receives an LL_REJECT_IND or LL_REJECT_EXT_IND PDU from the IUT, in the latter case specifying the opcode of the PDU sent in step 5.
   c. The Lower Tester receives a PDU from the IUT that would be a response to the opcode sent in step 5 with valid parameters.
   d. The Lower Tester does not receive an LL Control PDU from the IUT for 20 seconds (i.e., half the procedure response timeout).
7. If case (c) occurs in step 6, the Lower Tester completes the relevant procedure with the IUT.
8. If case (d) occurs in step 6, the Lower Tester sends an LL_PING_REQ to the IUT and receives an LL_PING_RSP or LL_UNKNOWN_RSP PDU in reply. If continuing the procedure leads to a breakdown of the connection (e.g., IUT and Lower Tester using different PHY, interval, or channel map), the Lower Tester may return an inconclusive verdict.

9. If the PDU sent in step 5 was the LL_TERMINATE_IND PDU and the IUT exits the connection state, the Upper and Lower Testers shall carry out the procedures necessary to re-establish the connection.

- **Expected Outcome**

  **Pass Verdict**
  - In step 6, the IUT behaves as specified in Table 4.98 based on the LL Control PDU sent by the Lower Tester.
  - If the IUT can process Invalid CtrData, the Lower Tester shall verify the IUT sends a valid response.
  - If the IUT rejects Invalid CtrData, the IUT shall send the Lower Tester an LL_UNKNOWN_RSP PDU, LL_REJECT_IND, or LL_REJECT_EXT_IND PDU specifying the correct opcode.

  **Inconclusive Verdict**
  - In step 8, if continuing the procedure leads to a breakdown of the connection (e.g., IUT and Lower Tester using different PHY, interval, or channel map), the Lower Tester may return an inconclusive verdict.

<table>
<thead>
<tr>
<th>LL Control PDU</th>
<th>Expected Results of Step 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL_CONNECTION_UPDATE_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_CHANNEL_MAP_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_TERMINATE_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_ENC_REQ</td>
<td>Option (a) or (b) or (c)</td>
</tr>
<tr>
<td>LL_ENC_RSP</td>
<td>Option (a) or (b) or (d)</td>
</tr>
<tr>
<td>LL_START_ENC_REQ</td>
<td>Option (a) or (b) or (d)</td>
</tr>
<tr>
<td>LL_START_ENC_RSP</td>
<td>Option (a) or (b) or (d)</td>
</tr>
<tr>
<td>LL_UNKNOWN_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_FEATURE_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_FEATURE_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_PAUSE_ENC_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_PAUSE_ENC_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_VERSION_IND</td>
<td>Option (a) or (c) or (d)</td>
</tr>
</tbody>
</table>
### Table 4.98: Expected Results of Step 6 for Section 4.7.4.1 “Control PDUs with Invalid Length”

<table>
<thead>
<tr>
<th>LL Control PDU</th>
<th>Expected Results of Step 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL_REJECT_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_SLAVE_FEATURE_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_CONNECTION_PARAM_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_CONNECTION_PARAM_RSP</td>
<td>Option (a) or (b) or (d)</td>
</tr>
<tr>
<td>LL_REJECT_EXT_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_PING_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_PING_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_LENGTH_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_LENGTH_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_PHY_REQ</td>
<td>Option (a) or (b) or (c)</td>
</tr>
<tr>
<td>LL_PHY_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_PHY_UPDATE_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_MIN_USED_CHANNELS_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_CTE_REQ</td>
<td>Option (a) or (b) or (c)</td>
</tr>
<tr>
<td>LL_CTE_RSP</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_PERIODIC_SYNC_IND</td>
<td>Option (a) or (d)</td>
</tr>
<tr>
<td>LL_CLOCK_ACCURACY_REQ</td>
<td>Option (a) or (c)</td>
</tr>
<tr>
<td>LL_CLOCK_ACCURACY_RSP</td>
<td>Option (a) or (d)</td>
</tr>
</tbody>
</table>

#### 4.8 SEC

Tests whether the IUT behaves according to the requirements for the security procedures. The security function specific abbreviations used are defined in [3].

#### 4.8.1 Common PDU Contents

The packet descriptions for Lower Tester sent and Lower Tester accepted data channel packets are below.
LL_ENC_REQ CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>LSO MSO</th>
<th>LSO MSO</th>
<th>LSO MSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
<td>lsb msb</td>
<td>lsb msb</td>
</tr>
<tr>
<td>Rand</td>
<td>EDIV</td>
<td>SKDm</td>
<td>IVM</td>
</tr>
<tr>
<td>(8 octets)</td>
<td>(2 octets)</td>
<td>(8 octets)</td>
<td>(4 octets)</td>
</tr>
</tbody>
</table>

LL_ENC_RSP CtrData:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>LSO MSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
</tr>
<tr>
<td>SKDs</td>
<td>IVs</td>
</tr>
<tr>
<td>(8 octets)</td>
<td>(4 octets)</td>
</tr>
</tbody>
</table>

Encrypted DATA PDU:

<table>
<thead>
<tr>
<th>LSO MSO</th>
<th>LSO MSO</th>
<th>MSO LSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
<td>msb lsb</td>
</tr>
<tr>
<td>Header</td>
<td>Payload</td>
<td>MIC</td>
</tr>
<tr>
<td>(2 octets)</td>
<td>(1-27 octets)</td>
<td>(4 octets)</td>
</tr>
</tbody>
</table>

4.8.2 ADV
Tests that the IUT behaves according to the security procedures in the advertiser role.

4.8.2.1 LL/SEC/ADV/BV-01-C [Advertising With Static Address]

- Test Purpose
  Test that an advertiser IUT is able to advertise using a static address and performs the scan response procedure with filtering settings allowing scan requests for devices supporting filtering.

  The Upper Tester generates a static address, assigns it to the IUT, and then the Lower Tester acts as an active scanner, observing the advertising packets and the static address included. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when advertising is enabled.

- Reference
  [3] 4.4.2

- Initial Condition
  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  State: Generate a static address AND Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, private own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser).
Test Procedure

Execute the test procedure using a static address as the advertiser address.

1. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT static address.
2. Upper Tester enables undirected advertising and sets scan response data in the IUT.
3. Configure the Lower Tester to start active scanning.
4. Lower Tester receives an ADV_IND packet from the IUT using the advertising static address.
5. Lower Tester responds with a SCAN_REQ packet using the Lower Tester static address and the IUT static address on the selected advertising channel T_IFS after the end of an advertising packet.
6. Lower Tester receives a SCAN_RSP packet from the IUT addressed to the Lower Tester after T_IFS from the end of the request packet.
7. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT static address and receives an HCI_Command_Complete event from the IUT with a Status of 0x0C.
8. Upper Tester sends an HCI_LE_Set_Advertising_Enable command to disable advertising in the IUT and receives an HCI_Command_Complete event from the IUT.

Figure 4.367: LL/SEC/ADV/BV-01-C [Advertising With Static Address]
• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a static address in advertising packets.

The IUT responds to the scan requests.

The IUT rejects the second HCI_LE_Set_Random_Address command.

4.8.2.2 LL/SEC/ADV/BV-02-C [Privacy - Non-connectable Undirected Advertising, non-resolvable private address]

• Test Purpose

Verify that an advertiser IUT is able to advertise non-connectable events using a non-resolvable private address.

The Upper Tester generates a non-resolvable private address, assigns it to the IUT. The Lower Tester then observes the packet contents on the advertising channel with the non-resolvable private address included.

• Reference

[3] 4.4.2.6

• Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Non-Connectable Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)

• Test Procedure

Execute the test procedure using a single non-resolvable private address as the advertiser address and a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packets AdvA field.
3. Upper Tester enables non-connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
4. Lower Tester expects the IUT to send ADV_NONCONN_IND packets on an applicable advertising channel.
5. Repeat steps 4 until 100 advertising events have been detected.
6. Upper Tester disables non-connectable advertising.
• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a non-resolvable private address and using it in non-connectable advertising packets.

The advertiser’s address received by the Lower Tester matches the address set by the Upper Tester.

4.8.2.3 LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address]

• Test Purpose

Verify that an advertiser IUT is able to advertise non-connectable events using a resolvable private address and that the address is refreshed.

The Lower Tester observes the packet contents on the advertising channel including the resolvable private address. The resolvable private address shall be refreshed within the Resolvable Private Address timeout.

• Reference

[3] 4.4.2.6

• Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Device Address Set (supported type of address, any address)

Execute the test procedure using a single local IRK
Test Procedure

Figure 4.369: LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address]
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertising packet's AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.
3. Upper Tester enables non-connectable advertising in the IUT using all supported advertising channels, a selected advertising interval between the minimum and maximum advertising intervals, and sets the Own_Address_Type to either 0x02 or 0x03 according to IUT address set in initial condition.
4. Lower Tester expects the IUT to send ADV_NONCONN_IND packets on an applicable advertising channel.
5. Lower Tester resolves private address received from the IUT using assigned IRK.
6. Lower Tester waits for address refresh timeout and resolves another ADV_NONCONN_IND packet verifying that a different address is used that resolves with the same IRK.
7. Upper Tester disables non-connectable advertising.

• Expected Outcome

Pass Verdict
The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in advertising and refreshing the address within the defined timeout period.

4.8.2.4 LL/SEC/ADV/BV-04-C [Network Privacy - Scannable Advertising, non-resolvable private address]

• Test Purpose
Verify that that an advertiser IUT is able to advertise scannable undirected events using a non-resolvable private address.

The Upper Tester generates a non-resolvable private address, assigns it to the IUT. The Lower Tester then observes the packet contents on the advertising channel with the non-resolvable private address included.

• Reference
[3] 4.4.2.5

• Initial Condition
Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map

State: Scannable undirected advertising (selected AdvInterval_Min, selected AdvInterval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)

Execute the test procedure using a single non-resolvable private address as the advertiser address.
1. Configure Lower Tester to active scan for advertising packets from the IUT.
2. Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
3. The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packet's AdvA field.
4. Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet.
packet from the IUT. The Lower Tester uses a non-resolvable private address in the SCAN_REQ packet.
5. Lower Tester expects the IUT to send SCAN_RSP packets in response to the SCAN_REQ.
6. Repeat steps 4–5 until 100 advertising events have been detected.
7. Upper Tester disables scannable undirected advertising.

• Expected Outcome

Pass Verdict
The test procedure is executed successfully, with the advertiser using a non-resolvable private address and using it in the AdvA field of scannable undirected advertising packets.

The advertiser address matches the address received by the Lower Tester.

4.8.2.5 LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address]

• Test Purpose

Verify that an advertiser IUT is able to advertise scannable undirected events using a resolvable private address. The resolvable private address shall be refreshed within the Resolvable Private Address timeout

The Lower Tester observes the packet contents on the advertising channel, and resolves the IUT’s resolvable private address. The Lower Tester then responds with a scan request using a resolvable private address and receives a scan response packet in return.

Verify that when the IUT’s address resolution is disabled, it no longer responds to the scan request.

• Reference

[3] 4.4.2.5

• Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)

Execute the test procedure using a single local IRK.
- **Test Procedure**

![Diagram of test procedure](image)

Figure 4.371: LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address]
1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
3. Upper Tester adds the Lower Tester’s Identity Address to the White List.
4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertising packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.
5. Upper Tester enables scannable undirected advertising with Advertising_Filter_Policy set to 0x01.
6. Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT.
8. IUT resolves the private address received from the Lower Tester using assigned IRK. Lower Tester expects the IUT to send SCAN_RSP packets in response to the SCAN_REQ.
9. Lower Tester waits for address refresh timeout and resolves another ADV_SCAN_IND packet verifying that a different address is used that resolves with the same IRK.
10. Upper Tester disables non-connectable advertising.
11. Upper Tester disables address resolution.
12. Upper Tester enables scannable undirected advertising.
13. Repeat steps 14–16 at least 20 times.
14. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
15. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT, setting the ScanA field to an address generated using its IRK.
16. IUT does not resolve the private address received from the Lower Tester. The Lower Tester does not receive SCAN_RSP packets in response to the SCAN_REQ.
17. Upper Tester disables the advertising.

• Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising and refreshing the address within the defined timeout period.

The Lower Tester and IUT successfully resolved the peer’s address and send the SCAN_REQ and SCAN_RSP, respectively.

IUT does not respond to the SCAN_REQ when address resolution is disabled.

4.8.2.6 LL/SEC/ADV/BV-06-C [Network Privacy - Undirected Connectable Advertising no Local IRK, no peer IRK]

• Test Purpose

Verify that an advertiser IUT can connect while using non-resolvable private address in the AdvA field.

• Reference

[3] 4.4.2.3
• Initial Condition

Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map

State: Undirected Advertising (selected AdvInterval_Min, selected AdvInterval_Max, supported type of own address, selected advertising channel map)

• Test Procedure

Configure Lower Tester to initiate a connection while using a Public Address.

The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packets AdvA field.

Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.

Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
5. Lower Tester connects to the IUT. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
6. Upper Tester terminates the connection.
7. Repeat steps 1–6 an additional 2 times, but configure the Lower Tester with a Static Address and Non-Resolvable Private Address, respectively.

- Expected Outcome
  
  Pass Verdict
  
The test procedure is executed successfully, with the advertiser using a non-resolvable private address in connectable advertising, connecting to the Lower Tester.

4.8.2.7 LL/SEC/ADV/BV-07-C [Network Privacy - Undirected Connectable Advertising with Local IRK, no peer IRK]

- Test Purpose
  
  Verify that an advertiser IUT can connect while using the Resolving List and using a resolvable private address in the AdvA field. The peer has not distributed its IRK. The Device Identity (IRK and Identity Address) of the IUT is known by the peer.
  
The Lower Tester has not distributed its own IRK and uses a Public or Static Address (Identity Address) for the InitA field, the identity address is stored in the IUT’s White List.

- Reference
  
  [3] 4.4.2.3

- Initial Condition
  
  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  
  State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)
• Test Procedure

1. Configure Lower Tester to initiate a connection while using the Identity Address.
2. Upper Tester adds the Lower Tester's Identity Address to the White List.
3. The Upper Tester adds the Lower Tester’s Device Identity (IRK and Identity Address) to the IUT resolving list.

4. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.

5. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.

6. Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA identical to its Identity Address.

7. The IUT verifies the AdvA and the InitA address of the CONNECT_IND. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

8. The Upper Tester terminates the connection.

- Expected Outcome

  Pass Verdict

  The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

  The IUT connects to the Lower Tester when the Lower Tester uses either a public or a static address.

4.8.2.8 LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK]

- Test Purpose

  Verify the IUT when transmitting undirected connectable advertising events, using the Resolving List and using a resolvable private address for AdvA field, connects to the Lower Tester. The Lower Tester uses a resolvable private address for the InitA field, i.e. the Lower Tester has distributed its own device identity (IRK and Identity Address) and uses resolvable private addresses. IUT and Lower Tester validate the address used towards the device identities.

  Verify that when address resolution is disabled on the IUT, the Lower Tester resolvable private address is not resolved, and therefore a connection is not established.

- Reference

  [3] 4.4.2.3

- Initial Condition

  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Undirected Advertising (selected Adv_Impl_Min, selected Adv_Impl_Max, supported type of own address, selected advertising channel map)
Test Procedure

- Test Procedure

Figure 4.374: LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK]
1. Configure Lower Tester to initiate a connection while using a resolvable private address.

2. The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK. The IUT uses this when generating a resolvable private address for use in the advertising packet's AdvA field.

3. Upper Tester adds the Lower Tester’s Identity Address to the White List.

4. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.

5. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.

6. Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its IRK. The IUT verifies the AdvA field, and resolves the address in the InitA field.

7. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

8. The Upper Tester terminates the connection.

9. Upper Tester disables address resolution in the IUT.

10. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.

11. Repeat steps 12–13 at least 20 times.

12. The Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel. The Lower Tester resolves the AdvA address and identifies the IUT.

13. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its IRK. The IUT does not resolve the address in the InitA field. No connection event is sent to the Upper Tester.

14. Upper Tester disables advertising on the IUT.

- Expected Outcome

**Pass Verdict**

The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

The Lower Tester also uses a resolvable private address during the connections and the IUT correctly resolves the address.

A connection is not established when address resolution is disabled.

4.8.2.9 LL/SEC/ADV/BV-09-C [Network Privacy - Undirected Connectable Advertising without Local IRK and with peer IRK]

- Test Purpose

Verify that the IUT connects to the Lower Tester when transmitting undirected connectable advertising events and using the Resolving List with a public or random static address for AdvA field. The Lower Tester has distributed its Device Identity (IRK and Identity Address) and uses a resolvable private address for the InitA field.

- Reference

[3] 4.4.2.3

- Initial Condition

Parameters: LLAdvertiser_advInterval_MIN, LLAdvertiser_advInterval_MAX, LLAdvertiser_Adv_Channel_Map

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address (0x02 or 0x03), selected advertising channel map)
The Upper Tester host has not provided an IRK for the IUT.

- **Test Procedure**

1. Configure Lower Tester to initiate a connection while using a resolvable private address.
2. Upper Tester adds the Lower Tester’s Identity Address to the White List.
3. The Upper Tester populates the resolving list with the device identity of the Lower Tester and a local IRK value of all zeros.
4. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The AdvA value is set to the IUT’s Identity Address.

5. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.

6. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its Device Identity. The IUT resolves the InitA Address and identifies the Lower Tester.

7. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

8. The Upper Tester terminates the connection.

• Expected Outcome

   Pass Verdict

   IUT advertises with a Public or Static address in the AdvA field.

   The IUT is able to use the resolving list to connect to the Lower Tester.

4.8.2.10 LL/SEC/ADV/BV-10-C [Network Privacy - Undirected Connectable Advertising using Resolving List and Peer Device Identity not in the List]

• Test Purpose

   Verify that the IUT, when transmitting undirected connectable advertising events and using the Resolving List, connects to the devices that are only resolved and on the White List. The IUT should only connect to the Lower Tester upon successful resolution of the peer’s resolvable private address.

• Reference

   [3] 4.4.2.3

• Initial Condition

   The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

   State: Undirected Advertising (selected Adv_Current_Min, selected Adv_Current_Max, supported type of own address, selected advertising channel map)
1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Upper Tester adds the Lower Tester’s Identity Address to the White List.
3. Configure the Lower Tester to initiate a connection while using a resolvable private address.
4. The Upper Tester populates the resolving list with device identities not equal to the one of the Lower Tester connected with the local device identity.
5. The IUT use its device identity when generating a resolvable private address for use in the advertising packet’s AdvA field.
6. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
7. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
8. Lower Tester sends a CONNECT_IND using the advertiser's address for AdvA field. The InitA field uses a resolvable private address based on the device identity of the Lower Tester.
9. The Lower Tester repeats step 7 ten times.

- Expected Outcome

**Pass Verdict**
- IUT advertises with a resolvable private address in the AdvA field.
- The IUT ignores the connect requests from addresses which are not in the resolving list.

4.8.2.11 LL/SEC/ADV/BV-11-C [Network Privacy - Directed Connectable Advertising using local and remote IRK]

- Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, is using resolvable private addresses for AdvA and InitA fields when the Lower Tester has distributed its own IRK.

Verify that when address resolution is disabled on the IUT, the Lower Tester resolvable private address is not resolved, and therefore a connection is not established.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

Lower Tester is using a resolvable private address and has previously distributed its IRK.

The IUT is using a resolvable private address.
Test Procedure

- Test Procedure

**Figure 4.377: LL/SEC/ADV/BV-11-C [Network Privacy - Directed Connectable Advertising using local and remote IRK]**
1. The Lower Tester adds the Device Identity of the IUT to its resolving list.

2. Configure Lower Tester to initiate a connection while using a resolvable private address.

3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity. The IUT use these when generating resolvable private addresses for use in the advertising packet’s AdvA and InitA fields.

4. Upper Tester enables resolving list and directed connectable advertising in the IUT.

5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

6. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_DIRECT_IND and the InitA generated based on its Device Identity. The IUT verifies AdvA and resolves the InitA Address and identifies the Lower Tester.

7. The Lower Tester connects to the IUT. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

8. The Upper Tester terminates the connection.

9. Upper Tester disables address resolution in the IUT.

10. Upper Tester enables directed connectable advertising in the IUT.

11. Repeat steps 12–13 at least 20 times.

12. The Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel. The Lower Tester resolves the AdvA address and identifies the IUT.

13. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its IRK. The IUT does not resolve the address in the InitA field. No connection event is sent to the Upper Tester.

• Expected Outcome

  Pass Verdict

IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT verifies the AdvA address and resolves the InitA address in the CONNECT_IND packet and accepts the connection.

A connection is not established when address resolution is disabled.

4.8.2.12 LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK]

• Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, is using resolvable private address for AdvA field and a Public or Static Address (Identity Address) for the InitA field when the Lower Tester has not distributed its own IRK.

• Reference

[3] 4.4.2.4

• Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT. Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)
The IUT is using a resolvable private address.
The Lower Tester is using a public or static address and has not distributed its IRK.

- **Test Procedure**

![Diagram](image-url)

*Figure 4.378: LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK]*
1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure Lower Tester to initiate a connection while using a Public Address.
3. The Upper Tester populates the resolving list with the identity address of the Lower Tester connected with the local device identity. The Lower Tester’s IRK is set to all zeros. The IUT uses these when generating resolvable private addresses for use in the advertising packet’s AdvA field.
4. Upper Tester enables directed connectable advertising in the IUT.
5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
6. The Lower Tester resolved the identity of the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA being its Public or Static Random Device Address. The IUT verifies the AdvA address and resolves the InitA address to identify the Lower Tester.
7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The Upper Tester terminates the connection.
9. The IUT has generated a new resolvable private address for the AdvA field.
10. Upper Tester enables connectable directed advertising in the IUT.
11. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

• Expected Outcome

Pass Verdict
IUT advertises with directed advertising using resolvable private address in the AdvA field.
The IUT accepts connection.
The IUT refreshes the advertiser address after the disconnection.

4.8.2.13 LL/SEC/ADV/BV-13-C [Network Privacy - Directed Connectable Advertising without local IRK but with remote IRK]

• Test Purpose
Verify the IUT when transmitting directed connectable advertising events using a public or static address for AdvA field and a resolvable private address for the InitA field when the Lower Tester has distributed its own IRK.

• Reference
[3] 4.4.2.4

• Initial Condition
The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT. Parameters: LL_advertiser_Adv_Channel_Map
State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)
The Lower Tester is using a resolvable private address and has distributed its IRK.
The IUT is using a public or static address.
**Test Procedure**

1. Configure Lower Tester to initiate a connection while using a resolvable private address.
2. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with a local all zero IRK. The IUT uses this when generating a resolvable private address for use in the advertising packet’s InitA field.

*Figure 4.379: LL/SEC/ADV/BV-13-C [Network Privacy - Directed Connectable Advertising without local IRK but with remote IRK]*
3. Upper Tester enables resolving list and directed connectable advertising in the IUT using Own_Address_Type to ResolvableOrPublic (0x02) or ResolvableOrRandom (0x03) according to IUT address (public or static random).

4. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel with initA as resolvable private address.

5. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA of the ADV_DIRECT_IND and an InitA address generated based on Lower Tester’s IRK. The IUT resolves the InitA Address and identifies the Lower Tester.

6. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

7. The Upper Tester terminates the connection and waits for the RPA Timeout.

8. The IUT has generated a new resolvable private address for the InitA and AdvA field.

9. Upper Tester enables connectable directed advertising in the IUT.

10. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

• Expected Outcome

Pass Verdict

IUT advertises with directed advertising using identity address in the AdvA and RPA in the initA address.

The IUT accepts the connection.

The IUT refreshes the initiator address after the private address refresh timeout.


• Test Purpose

Verify the IUT when transmitting directed connectable advertising events and using the Resolving List connects to the devices that are only in the resolving list. The IUT should only connect to a Lower Tester upon successful resolution of the peer’s resolvable private address.

• Reference

[3] 4.4.2.4

• Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LLAdvertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

The Lower Tester is using a resolvable private address and has distributed its IRK.

The IUT is using a public or static address.
1. Configure Lower Tester to initiate a connection while using a Resolvable Private Address.
2. The Upper Tester populates the resolving list with device identities not equal to the one of the Lower Tester connected with the local device identity.
3. The IUT use its device identity when generating a resolvable private address for use in the advertising packet’s AdvA field and the device identity in the resolving list for the InitA field.
4. Upper Tester enables directed connectable advertising in the IUT.
5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

Figure 4.380: LL/SEC/ADV/BV-14-C [Network Privacy - Directed Connectable Advertising using Resolving List and Peer Device Identity not in the List]
6. Lower Tester sends a CONNECT_IND using the advertiser's address for AdvA field. The InitA field uses a resolvable private address based on the device identity of the Lower Tester (an IRK that is not the same as the one distributed to the Upper Tester).

- Expected Outcome

**Pass Verdict**

IUT advertises with a resolvable private address in the AdvA field.

The IUT ignores the connect requests from addresses which are not in the resolving list.

IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT resolves the InitA address in the CONNECT_IND packet and does not accept the connection.

### 4.8.2.15 LL/SEC/ADV/BV-15-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore Identity Address]

- **Test Purpose**

Verify that an advertiser IUT does not respond to a scan request with the scanner's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

- **Reference**

[3] 4.4.2.5

- **Initial Condition**

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)

Execute the test procedure using a single local IRK.

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.
Test Procedure

1. Configure the Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The Advertising_Filter_Policy shall be set to 0x01 (Process connection requests from all devices and only scan requests from devices that are in the White List).
3. The Upper Tester adds the Lower Tester’s Identity Address to the White List.
4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.
5. The Upper Tester enables scannable undirected advertising.
6. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.

7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's device identity address.

8. The IUT compares the address by checking against its resolving list and finds a match with network privacy mode.

9. The Lower Tester expects that the IUT does not send SCAN_RSP packets in response to the SCAN_REQ.

10. The Upper Tester disables advertising.

- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

The Lower Tester and IUT successfully resolved the peer's address.

The IUT does not respond to the SCAN_REQ with the Lower Tester's device identity address.

4.8.2.16 LL/SEC/ADV/BV-16-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Ignore Identity Address]

- Test Purpose

Verify that the IUT, when transmitting undirected connectable advertising events, does not connect in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

- Reference

[3] 4.4.2.3

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.
• Test Procedure

1. Configure Lower Tester to initiate a connection while using its device identity address.
2. The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK. The IUT uses this when generating a resolvable private address for use in the advertisement packet’s AdvA field.
3. The Upper Tester adds the Lower Tester’s Identity Address to the White List.
4. The Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
5. The Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA set to its device identity address.
7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The Lower Tester expects the IUT to ignore the CONNECT_IND and expects the IUT not to maintain the connection.
9. The Upper Tester disables advertising.

• **Expected Outcome**

**Pass Verdict**

The test procedure is executed successfully, with the advertiser generating a resolvable private address and using it in connectable advertising.

The IUT ignores the CONNECT_IND and does not maintain the connection.

4.8.2.17 LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address]

• **Test Purpose**

Verify that the IUT, when transmitting directed connectable advertising events, does not connect in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

• **Reference**

[3] 4.4.2.4

• **Initial Condition**

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

Lower Tester is using its identity and has previously distributed its IRK.

The IUT is using a resolvable private address.

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.
Test Procedure

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure Lower Tester to initiate a connection while using its device identity address.
3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity. The IUT uses this when generating resolvable private addresses for use in the advertisement packet's AdvA and InitA fields.
4. The Upper Tester enables resolving list and directed connectable advertising in the IUT.
5. The Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_DIRECT_IND and the InitA set to its device identity address.
7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

Figure 4.383: LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address]
8. The Lower Tester expects the IUT to ignore the CONNECT_IND and expects the IUT not to maintain the connection.

9. The Upper Tester disables advertising.

• Expected Outcome
  
  **Pass Verdict**
  
  The IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.
  
  The IUT ignores the CONNECT_IND and does not maintain the connection.

4.8.2.18 LL/SEC/ADV/BV-18-C [Device Privacy - Scannable Advertising, resolvable private address, Accept Identity Address]

• Test Purpose
  
  Verify that an advertiser IUT responds to a scan request with the scanner’s identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

• Reference
  
  [3] 4.4.2.5

• Initial Condition
  
  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
  
  State: Scannable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)
  
  Execute the test procedure using a single local IRK.
1. Configure the Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
3. The Upper Tester adds the Lower Tester’s Identity Address to the White List.
4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.

5. The Upper Tester enables scannable undirected advertising.

6. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.

7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's device identity address.

8. The IUT compares the address by checking against its resolving list and finds a match with device privacy mode.

9. The Lower Tester expects the IUT to send a SCAN_RSP packet in response to the SCAN_REQ.

10. The Upper Tester disables advertising.

- **Expected Outcome**

  **Pass Verdict**

  The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

  The IUT responds to the SCAN_REQ with the advertiser RPA.

---

**4.8.2.19 LL/SEC/ADV/BV-19-C [Device Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Accept Identity Address]**

- **Test Purpose**

  Verify that the IUT, when transmitting undirected connectable advertising events, connects in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

- **Reference**

  [3] 4.4.2.3

- **Initial Condition**

  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IUT.

  **Parameters:** LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  **State:** Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)
Test Procedure

1. Configure the Lower Tester to initiate a connection while using its device identity address.
2. The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK and sets the entry to device privacy mode. The IUT uses this when generating a resolvable private address for use in the advertisement packet's AdvA field.
3. The Upper Tester adds the Lower Tester's Identity Address to the White List.
4. The Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.

5. The Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.

6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA set to its device identity address. The IUT verifies the AdvA field and compares the InitA address by checking against its resolving list and finds a match with device privacy mode.

7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

8. The connection is maintained for a time interval greater than or equal to the timeout value for the private address refresh.

9. The IUT terminates the connection.

- Expected Outcome

  Pass Verdict

  The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

4.8.2.20  LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address]

- Test Purpose

  Verify that the IUT, when transmitting directed connectable advertising events, connects in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

- Reference

  [3] 4.4.2.4

- Initial Condition

  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT. Parameters: LL_advertiser_Adv_Channel_Map

  State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

  Lower Tester is using its identity address and has previously distributed its IRK.

  The IUT is using a resolvable private address.
Test Procedure

Figure 4.386: LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address]
1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure the Lower Tester to initiate a connection while using its device identity address.
3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity and sets the entry to device privacy mode. The IUT uses this when generating resolvable private addresses for use in the advertisement packet’s AdvA and InitA fields.
4. The Upper Tester enables resolving list and directed connectable advertising in the IUT.
5. The Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
6. The Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_DIRECT_IND and the InitA set to its device identity address.
7. The Lower Tester connects to the IUT. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The IUT terminates the connection.

- **Expected Outcome**

  **Pass Verdict**

  IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

  The IUT verifies the AdvA address and compares the InitA address in the CONNECT_IND packet by checking against its resolving list and finds a match with device privacy mode and accepts the connection.

4.8.2.21 LL/SEC/ADV/BV-21-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA]

- **Test Purpose**

  Verify that an advertiser IUT does not respond to a scan request with the scanner’s RPA when the IUT has that address and an associated IRK in the resolving list using network privacy mode but address resolution is not enabled.

- **Reference**

  [3] 4.4.2.5

- **Initial Condition**

  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

  Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

  State: Scannable Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)

  Execute the test procedure using a single local IRK.
• **Test Procedure**

**Figure 4.387: LL/SEC/ADV/BV-21-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA]**

1. Configure the Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester assigns a valid Local IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field using add device to resolving list command.
3. The Upper Tester adds the Lower Tester's Identity Address to the White List.
4. The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The Advertising_Filter_Policy shall be set to 0x03 (Process scan requests and connect requests from devices that are in the White List).
5. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
6. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's RPA.
7. The IUT compares the lower tester’s address and does not try to resolve the lower tester's RPA.
8. The Lower Tester expects that the IUT does not send SCAN_RSP packets in response to the SCAN_REQ.
9. The Upper Tester disables advertising.

• **Expected Outcome**

**Pass Verdict**

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

The IUT does not respond to the SCAN_REQ with the Lower Tester’s RPA.
4.8.2.22 LL/SEC/ADV/BV-22-C [Network Privacy – Directed Connectable Advertising using Target RPA as InitA]

- Test Purpose
  Verify the IUT when transmitting directed connectable advertising events using a resolvable private address (RPA) for the InitA field connects when the LT uses the same InitA RPA in the CONNECT_IND.

- Reference
  [3] Sections 4.4.2.4, 6.4

- Initial Condition
  The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.
  Parameters: LLAdvertiser_Adv_Channel_Map
  State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)
  The Lower Tester is using a resolvable private address and has distributed its IRK.
  The IUT is using a public or static address.
1. Configure Lower Tester to initiate a connection while using a resolvable private address.

2. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with a local all zero IRK. The IUT uses this when generating a resolvable private address for use in the advertising packet’s InitA field.

Figure 4.388: LL/SEC/ADV/BV-22-C [Network Privacy – Directed Connectable Advertising using Target RPA as InitA]
3. Upper Tester enables resolving list and directed connectable advertising in the IUT using Own_Address_Type to ResolvableOrPublic (0x02) or ResolvableOrRandom (0x03) according to IUT address (public or static random).

4. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel with initA as resolvable private address.

5. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA and InitA of the ADV_DIRECT_IND. The IUT resolves the InitA Address and identifies the Lower Tester.

6. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.

7. The Upper Tester terminates the connection.

- Expected Outcome
  
  **Pass Verdict**

  The IUT advertises with directed advertising using identity address in the AdvA and RPA in the InitA address.

  The IUT accepts the connection with the CONNECT_IND InitA matching the ADV_DIRECT_IND InitA.

### 4.8.3 SCN

Tests that the IUT handles random addresses in the scanner role.

#### 4.8.3.1 LL/SEC/SCN/BV-01-C [Random Address Scanning]

- **Test Purpose**

  Test that a scanner IUT is able to calculate and use random addresses. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when scanning is enabled.

- **Reference**

  [3] 5.1.3, 4.4.3.2

- **Initial Condition**

  State: Random Address Calculated (common ir) AND Active Scanning (random address, selected scan interval, selected scan window) AND All White Listed (policy for scanner)

- **Test Procedure**

  Execute the test procedure using a single random address as the advertiser address. The test uses the common variable ‘ir’ to encrypt and resolve random addresses. The preambles steps calculate the identity resolving key (variable ‘irk’).
1. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT random address.
2. Upper Tester enables active scanning in the IUT.
3. Configure the Lower Tester the advertising with random device address.
4. Lower Tester sends an ADV_IND packet with the private address, each advertising event on the selected advertising channel only, using the selected advertising interval.
5. Lower Tester receives an SCAN_REQ with the Lower Tester random address and the IUT random address in response after T_IFS to any of the packets.
6. Lower Tester sends an SCAN_RSP containing the random address to the IUT T_IFS after of the response.
7. Upper Tester receives an HCI_LE_Advertising_Report event containing the advertising packet information.
8. Upper Tester receives an HCI_LE_Advertising_Report event from the IUT containing the address in the response sent in step 6.
9. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT random address and receives an HCI_Command_Complete event from the IUT with a Status of 0x0C.
10. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and receives an HCI_Command_Complete event in response.
11. Upper Tester sends an HCI_LE_Encrypt command to the IUT with the parameters 'irk' and 3 least significant octets of the advertiser address received. Receive an HCI_Command_Complete event in response with the result parameter matching to the 3 most significant octets of the advertiser address received.

• Expected Outcome
  Pass Verdict
  The IUT generates a random address.
  The IUT uses the random address in the SCAN_REQ packets.
  The IUT resolves a random address received in the ADV_IND packets.
  The IUT rejects the second HCI_LE_Set_Random_Address command.

4.8.4 INI
Tests that the IUT behaves according to the security procedures in the initiator role.

4.8.5 SLA
Tests that the IUT behaves according to the security procedures in the slave role.

4.8.5.1 LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup]

• Test Purpose
  Test that a slave IUT can perform the encryption mode change procedure, using the correct encryption.
  The Lower Tester acts as a master, maintaining a connection, then initiates the encryption mode setup.

• Reference
  [3] 5.1.3

• Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)
  State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)

• Test Procedure
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (common variables 'er' and 'ir').
The test procedure uses the common variables ‘ir’ and ‘er’ to calculate a diversifier hiding key (‘dhk’), to derive an encrypted diversifier (‘ediv’), to check the diversifier and finally to derive a long term key (‘ltk’) to set to the slave device.

**Figure 4.390: LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup]**

1. In the preamble steps (connection establishment use public address type for the Lower Tester and a supported type of address for the IUT.
2. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
3. Lower Tester sends empty DATA packets once a connection interval to the IUT and receives a DATA packet in response from the IUT until all data sent in 2 have been received.
4. In the Lower Tester, calculate the master portions of the session key diversifier, the initialization vector and a random number.
5. Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT, receives an LL_ENC_RSP in response. Continue the master transmissions.
6. Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and ‘ediv’ parameters sent in step 5.

7. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply with ‘ltk’ to the IUT and receives an HCI_Command_Complete in response.

8. Lower Tester receives an LL_START_ENC_REQ packet in response to the master transmissions and acknowledges this packet.

9. Lower Tester sends a LL_START_ENC_RSP packet encrypted to the IUT and receives an LL_START_ENC_RSP packet encrypted in response.

10. Upper Tester receives an HCI_Encryption_Enable event with encryption enable set to on.

11. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.

12. Lower Tester sends empty DATA packet once a connection interval to the IUT and expects the DATA packets from the IUT to contain encrypted payloads with matching MIC fields until all data sent in step 11 has been received.

13. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

14. In the preamble steps (connection establishment use random address type for the Lower Tester and a random type of address for the IUT.

15. Repeat steps 2–13.

• Expected Outcome

Pass Verdict

The test procedure executes successfully, with the data correctly transmitted and reported by the IUT.

The IUT sends its initialization vector and session key diversifier in a LL_ENC_RSP packet.

The IUT reports the encryption mode setup requested with the HCI event.

The IUT sends a LL_START_ENC_REQ packet until acknowledged.

The IUT acknowledges the LL_START_ENC_RSP and responds with one.

The IUT reports the encryption mode setup with the HCI event.

4.8.5.2 LL/SEC/SLA/BV-02-C [Slave Pause Encryption]

• Test Purpose

Test that a slave IUT can perform the encryption pause procedure.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption pause procedure.

• Reference

[3] 5.1.3.2

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)
• **Test Procedure**

1. As preamble: an encrypted connection is established (IUT acts as slave).
2. Lower Tester sends an encrypted LL_PAUSE_ENC_REQ packet and receives an encrypted LL_PAUSE_ENC_RSP packet in response.
3. When the Lower Tester receives the LL_PAUSE_ENC_RSP packet, it responds with an unencrypted LL_PAUSE_ENC_RSP. The connection is not encrypted at this point.
4. Lower Tester sends an LL_ENC_REQ packet to re-enable encryption using a new session key and receives an LL_ENC_RSP packet in response.
5. Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 4.
6. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply with 'ltk' to the IUT and receives an HCI_Command_Complete in response.
7. Lower Tester receives an LL_START_ENC_REQ in response to the master transmissions and acknowledges the packet.
8. Lower Tester sends an LL_START_ENC_RSP packet encrypted to the IUT and receives an LL_START_ENC_RSP packet encrypted in response.

*Figure 4.391: LL/SEC/SLA/BV-02-C [Slave Pause Encryption]*
9. Upper Tester receives an HCI_Encryption_Key_Refresh_Complete_Event event with encryption enable set to on.
10. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

- Expected Outcome

  Pass Verdict
  The IUT pauses encryption.
  The IUT sends LL_START_ENC_REQ packet.
  The IUT sends HCI_Encryption_Key_Refresh_Complete_Event once encryption is resumed.

4.8.5.3 LL/SEC/SLA/BV-03-C [Slave Pause Encryption Sending Data]

- Test Purpose
  Test that a slave IUT does not send any data packet during the encryption pause procedure.
  The Lower Tester acts as a master, maintaining a connection, then initiates the encryption pause procedure and submits data for the slave to transmit. The IUT should not send the data packet until encryption is resumed.

- Reference
  [3] 5.1.3.2

- Initial Condition
  State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value) State for executing with encrypted private addresses: Encrypted Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).
Test Procedure

1. As preamble: an encrypted connection is established (IUT acts as slave).
2. Lower Tester sends an encrypted LL_PAUSE_ENC_REQ packet and receives an encrypted LL_PAUSE_ENC_RSP packet in response.
3. When Lower Tester receives the LL_PAUSE_ENC_RSP packet, responds unencrypted LL_PAUSE_ENC_RSP and now the connection is not encrypted.
4. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and one data elements with the value 0xFF.
5. Lower Tester sends an LL_ENC_REQ packet to re-enable encryption using a new session key and receives an LL_ENC_RSP packet in response.
6. Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and ‘ediv’ parameters sent in step 4.
7. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply command with 'ltk' to the IUT and receives an HCI_Command_Complete in response.
8. Lower Tester receives an LL_START_ENC_REQ packet in response to the master transmissions. Acknowledge the packet.
9. Lower Tester sends an LL_START_ENC_RSP packet encrypted to the IUT and receives an LL_START_ENC_RSP packet encrypted in response.
10. Upper Tester receives an HCI_Encryption_Key_Refresh_Complete event with encryption enable set to on.
11. Lower Tester sends an empty DATA packet once a connection interval to the IUT and expects the DATA packet from the IUT to contain an encrypted payload with matching MIC field.
12. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

• Expected Outcome
  
  **Pass Verdict**
  
  The IUT sends the data packet once encryption is resumed.

4.8.5.4 LL/SEC/SLA/BV-04-C [Slave Sending LL_REJECT_IND]

• Test Purpose
  
  Test that, while executing the Encryption Start Procedure, a slave IUT sends a correct LL_REJECT_IND control packet if the Host does not provide a Long Term Key.
  
  The Lower Tester acts as master, maintaining a connection, then initiates the encryption procedure, failing to provide a Long Term Key to the slave IUT.

• Reference
  
  [3] 5.1.3.1

• Initial Condition
  
  State: Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).
  
  Lower Tester does not support LL_REJECT_EXT_IND.

• Test Procedure
  
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).
1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.

2. Upper Tester sends an HCI_LE_Encrypt command to the IUT, with the key ‘ir’ and number 2 and receives an HCI_Command_Complete containing the ‘dhk’ in response.

3. Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT. Receive an LL_ENC_RSP packet in response and continue the master transmissions.

4. Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and ‘ediv’ parameters sent in step 3.

5. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Negative_Reply command to the IUT and receives an HCI_Command_Complete in response.

6. Lower Tester receives an LL_REJECT_IND packet from the IUT with ErrorCode indicating “PIN or Key missing”.

- Expected Outcome

  **Pass Verdict**

  Lower Tester receives an LL_ENC_RSP packet in response to the LL_ENC_REQ packet it had previously sent to the IUT.

  Upper Tester receives an HCI_Command_Complete event with the Status parameter set to ‘Success’ in response to the HCI_LE_Long_Term_Key_Requested_Negative_Reply command.

  Lower Tester receives an LL_REJECT_IND packet from the IUT with ErrorCode indicating ‘PIN or Key missing’.

4.8.5.5 LL/SEC/SLA/BV-05-C [Slave Receiving Encrypted Data]

- Test Purpose

  Test that a slave IUT can receive encrypted data.

  The Lower Tester acts as a master, maintaining an encrypted connection, then initiates sending encrypted data packets without fragmentation.
• Reference
[3] 5.1.3

• Initial Condition

State: Connected Slave (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)

• Test Procedure

1. Configure Lower Tester to send 1000 encrypted data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
2. Lower Tester sends a DATA packet once a connection interval to the IUT, using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00. Lower Tester receives an empty DATA packet in response from the IUT.
3. Upper Tester receives an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 1 with the Packet_Boundary_Flag flag set to 0x02.
4. Repeat steps 2–3 until all data sent in step 1 has been reported.

• Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent,
The IUT reports all data correctly with HCI_LE_Data_Packet events using the HCI fragmentation flags, as specified in Section 4.1.6.

4.8.5.6 LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled]

• Test Purpose

Verify that the IUT as slave sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LEAuthenticated Payload Timeout,
in order to force the Lower Tester to transmit an LE ACL packet (LL_PING_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

- **Reference**
  
  [3] 5.1.8

- **Initial Conditions**
  
  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

  The connection is encrypted.

  The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.

- **Test Procedure**

  ![Diagram](image)

  **Figure 4.395: LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled]**

  1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_PING_RSP.
  2. The Lower Tester responds with LL_PING_RSP.

- **Expected Outcome**

  **Pass Verdict**

  The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.
• Notes
The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.5.7  LL/SEC/SLA/BV-07-C [Responding to LL_PING_REQ]

• Test Purpose
Verify that the IUT as slave responds to an LL_PING_REQ sent by the Lower Tester.

• Reference
[3] 5.1.8

• Initial Condition
State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)
The connection is encrypted.
The connection is kept idle i.e., no LE-U or LE-C traffic is exchanged.

• Test Procedure

1. The Lower Tester transmits the PDU LL_PING_REQ.
2. The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

• Expected Outcome
Pass Verdict
The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

4.8.5.8  LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ]

• Test Purpose
Verify that the IUT as slave generates the HCI Authenticated Payload Timeout Expired event when the Lower Tester doesn’t send a packet containing a valid MIC to the IUT within the Authenticated_Payload_Timeout interval.
• Reference

[3] 5.1.8

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

• Test Procedure

Figure 4.397: LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ]

1. The Upper Tester sets the Authenticated_Payload_Timeout (defined as an IXIT).
2. The Upper Tester unmasks the HCI Authenticated Payload Timeout Expired event.
3. The Lower Tester sends a data packet containing valid MIC.
4. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.
5. The IUT transmits the PDU LL_PING_REQ to the Lower Tester.
6. The Lower Tester does not respond with LL_PING_RSP.
7. The IUT sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester
Authenticated_Payload_Timeout (defined as an IXIT) after the last packet that contained a MIC was received by the IUT from the Lower Tester.
• Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to the Lower Tester and sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester when the Lower Tester doesn’t respond with an LL_PING_RSP.

4.8.5.9 LL/SEC/SLA/BV-09-C [Modified Authentication Payload Timeout]

• Test Purpose

Verify that the IUT as slave uses the correct value of the Authenticated Payload Timeout (greater than 100 s or less than 5 s) set by the Upper Tester.

• Reference

[3] 5.1.8

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

• Test Procedure

1. The Upper Tester writes Authenticated_Payload_Timeout (defined as an IXIT) less than 5 s.
2. The Lower Tester sends a data packet containing valid MIC.
3. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged for time greater than Authenticated_Payload_Timeout (defined as an IXIT).
4. The IUT transmits the PDU LL_PING_REQ before time which is less than Authenticated_Payload_Timeout (defined as an IXIT).

5. The Lower Tester responds with LL_PING_RSP. The time between the two packets from the Lower Tester containing a MIC shall not be greater than Authenticated_Payload_Timeout (defined as an IXIT).

• Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ and receives the resulting LL_PING_RSP PDU within the time defined in the IXIT for Authenticated_Payload_Timeout after receiving a packet containing a valid MIC from the Lower Tester.

• Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another Authenticated_Payload_Timeout time.

4.8.5.10 LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

• Test Purpose

Verify that the IUT as slave sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_UNKNOWN_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

• Reference

[3] 5.1.8

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

Lower Tester does not support LE Ping Procedure.

The connection is encrypted.

The connection is kept idle i.e., no LE-U or LE-C traffic is exchanged.
• **Test Procedure**

![Diagram showing an encrypted connection between IUT and Lower Tester](image)

**Figure 4.399:** LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_UNKNOWN_RSP.
2. The Lower Tester responds with LL_UNKNOWN_RSP.

• **Expected Outcome**

**Pass Verdict**

The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.

• **Notes**

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

### 4.8.5.11 LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND]

• **Test Purpose**

Test that, while executing the Encryption Start Procedure, a slave IUT sends a correct LL_REJECT_IND or LL_REJECT_EXT_IND control packet if the Host does not provide a Long Term Key.

The Lower Tester acts as master, maintaining a connection, then initiates the encryption procedure and the Upper Tester fails to provide a Long Term Key to the slave IUT.

• **Reference**

[3] 5.1.3.1
- **Initial Condition**

  Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

  State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- **Test Procedure**

  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).

  ![Diagram](image)

  *Figure 4.400: LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND]*

1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT. The Lower Tester initiates a features exchange procedure.

2. Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT. Receive an LL_ENC_RSP packet in response and continue the master transmissions.

3. Upper Tester receives an HCI_LE_Long_Term_Key-Requested event from the IUT, containing the random number and ‘ediv’ parameters sent in step 3.

4. Upper Tester sends an HCI_LE_Long_Term_Key-Requested_Negative_Reply command to the IUT and receives an HCI_Command_Complete in response.

5. Lower Tester receives an LL_REJECT_IND or LL_REJECT_EXT_IND PDU from the IUT with the RejectOpcode set to “LL_ENC_REQ” and ErrorCode indicating “PIN or Key missing”.
• Expected Outcome

Pass Verdict

Lower Tester receives an LL_ENC_RSP packet in response to the LL_ENC_REQ packet it had previously sent to the IUT.

Upper Tester receives an HCI_Command_Complete event with the Status parameter set to ‘Success’ in response to the HCI_LE_Long_Term_Key_Requested_Negative_Reply command.

Lower Tester receives an LL_REJECT_IND or LL_REJECT_EXT_IND PDU from the IUT with the RejectOpcode set to “LL_ENC_REQ” and ErrorCode indicating “PIN or Key missing”.

4.8.5.12 LL/SEC/SLA/BI-01-C [Slave Encryption Setup: Missing Response]

• Test Purpose

Test that a slave IUT can perform the encryption mode change procedure and recover from a master device failing to send a start encryption response packet.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption mode setup, failing before the setup is complete.

• Reference

[3] 5.1.3

• Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).

• Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).
1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
2. Lower Tester, calculates the master portions of the session key diversifier, the initialization vector and a random number.
3. Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT. Receive an LL_ENC_RSP packet in response and continue the master transmissions.
4. Upper Tester receives an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and ‘ediv’ parameters sent in step 3.
5. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply command with ‘ltk’ to the IUT and receives an HCI_Command_Complete in response.
7. Lower Tester continues the master transmissions, but does not send a LL_START_ENC_RSP packet to the IUT.
8. Lower Tester expects the IUT to stop slave transmissions when the connection control timer expires.
9. Interleave with step 8: Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination from LL Response Timeout (0x22).
• Expected Outcome
Pass Verdict
The test procedure executes successfully, with the IUT stopping slave transmissions,
The IUT reports the termination of the connection with an HCI event.

4.8.5.13 LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC]

• Test Purpose
Test that a slave IUT terminates an encrypted connection upon a MIC failure in a packet received from the master device.
The Lower Tester acts in the master role, maintaining an encrypted connection and transferring data, then causes connection termination by corrupting the data packet contents.

• Reference
[3] 3.1, 5.1.3

• Initial Condition
State: Encrypted Slave Connection (common identity root, common encryption root, any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

• Test Procedure

![Diagram](image)

Figure 4.402: LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC]

1. Preamble: starting from an encrypted connection setup using the common connection parameters and encryption key variables.
2. Lower Tester sends a DATA packet with 10 bytes of data, flipping a single bit in the MIC after calculation, but using a correct CRC for the invalid encrypted packet.
3. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
4. Repeat step 2 receiving no response from the IUT up to a time equal to the connection supervision timeout value.

- Expected Outcome
  
  Pass Verdict
  
  The test procedure executes successfully, with the IUT terminating the connection after having tried to authenticate a corrupted data packet.
  
  The IUT does not report to the Upper Tester any of the corrupted data packets received from the Lower Tester.
  
  The IUT reports the connection termination with an HCI event.

4.8.5.14 LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header]

- Test Purpose
  
  Test that a slave IUT terminates an encrypted connection upon a MIC failure in a packet received from the master device.
  
  The Lower Tester acts in the master role, maintaining an encrypted connection and transferring data, then causes connection termination by corrupting the data packet contents.

- Reference
  
  [3] 3.1, 5.1.3

- Initial Condition
  
  State: Encrypted Slave Connection (common identity root, common encryption root, any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure
  
  Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.

Figure 4.403: LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header]
1. Lower Tester sends a DATA packet with 10 bytes of data, flipping a single bit in the header (LLID or RFU) after MIC calculation, but using a correct CRC for the invalid encrypted packet.
2. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
3. Repeat step 1 receiving no response from the IUT, up to a time equal to the connection supervision timeout value.

- Expected Outcome
  
  **Pass Verdict**

  The test procedure executes successfully, with the IUT terminating the connection after having tried to authenticate a corrupted data packet received from the Lower Tester.

  The IUT does not report to the Upper Tester any of the corrupted data packets received from the Lower Tester.

  The IUT reports the connection termination with an HCI event.

4.8.5.15 LL/SEC/SLA/BI-05-C [Slave Receiving unexpected PDU during encryption start]

- Test Purpose

  Test that a slave IUT which has started the encryption procedure does not respond to an LL_VERSION_IND but instead drops the link.

  The Lower Tester acts as a master.

- Reference

  [3] 5.1.3.1

- Initial Condition

  State: Connected Slave (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).
1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
2. Lower Tester calculates the master portions of the session key diversifier, the initialization vector and a random number.
3. Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT.
4. The LL_ENC_REQ packet is immediately followed by a LL_VERSION_IND packet from the Lower Tester. (In the same connection event if possible).
5. [Optional] IUT may send LL_ENC_RSP.
6. The IUT should terminate the connection on receiving the LL_VERSION_IND packet.
7. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination from “Connection Terminated Due to MIC Failure (0x3D).

- Expected Outcome

Pass verdict:
The IUT terminates the connection upon receiving the unexpected LL_VERSION_IND packet.

### 4.8.6 MAS

Tests that the IUT behaves according to the security procedures in the master role.

#### 4.8.6.1 LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup]

- **Test Purpose**
  
  Test that a master IUT is able to request encryption mode setup from a slave Controller, correctly encrypting the packets and performing the procedure.

  The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT.
**Initial Condition**

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout)

**Test Procedure**

Execute the test procedure with the common connection parameters and using the common variables for the encryption roots (variables ‘er’, and ‘ir’).

---

**Figure 4.405: LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup]**

1. In the preamble steps: connection establishment use public address type for the Lower Tester and a supported type of address for the IUT.
2. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
3. Lower Tester receives DATA packets from the IUT, including the data sent in step 2 and sends empty DATA packets to the IUT until all data sent in 2 has been received.
4. Lower Tester, calculates the slave portions of the session key diversifier, the initialization vector and a random number.
5. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution and ‘ltk’. Receive an HCI_Command_Status event in response.

6. Lower Tester receives an LL_ENC_REQ packet including the parameters from 5: with the master portions of the session key diversifier and the initialization vector. Acknowledge the encryption request and respond with an LL_ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.

7. Configure the Lower Tester calculating the session key diversifier and initialization vector from the parameters exchanged in step 6. Calculate the diversifier, using ‘dhk’ and the parameters received in step 6. Calculate the ‘ltk’ from the diversifier using ‘er’. Check the ltk to match the calculations in the preamble steps. Calculate the session key using the session key diversifier.

8. Lower Tester receives an empty DATA packet from the IUT and responds with an LL_START_ENC_REQ packet.

9. Lower Tester expects an LL_START_ENC_RSP packet encrypted from the IUT and responds with an LL_START_ENC_RSP packet encrypted.

10. Upper Tester receives an HCI_Encryption_Change event from the IUT, containing the connection handle from the preamble steps’ execution.

11. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.

12. Lower Tester expects the DATA packets received from the IUT to be encrypted with matching MIC fields and sends empty DATA packets in response until all data sent in step 2 have been received.

13. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

14. In the preamble steps (connection establishment use random address type for the Lower Tester and a random type of address for the IUT).

15. Repeat steps 2–13.

• Expected Outcome

Pass Verdict
The test procedure executes successfully, with the data transmitted correctly reported by the IUT. The IUT sends its initialization vector and session key diversifier in an LL_ENC_REQ packet.

4.8.6.2 LL/SEC/MAS/BV-02-C [Master Pause Encryption]

• Test Purpose
Test that a master IUT can perform the encryption pause procedure.

The Lower Tester acts in the slave role, accepts the encryption pause request from the IUT.

• Reference
[3] 5.1.3.2
• **Initial Condition**
  
  State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout)

• **Test Procedure**

![Diagram of test procedure](image)

*Figure 4.406: LL/SEC/MAS/BV-02-C [Master Pause Encryption]*

1. As preamble: an encrypted connection is established (IUT acts as master).
2. Upper Tester sends an HCI_LE_Start_Encryption and receives an HCI_Command_Status in response.
3. Lower Tester receives an encrypted LL_PAUSE_ENC_REQ packet and sends an encrypted LL_PAUSE_ENC_RSP packet in response.
4. Lower Tester receives an unencrypted LL_PAUSE_ENC_RSP packet from the IUT. At this point the connection stops being encrypted.
5. Lower Tester receives an ENC_REQ packet to re-enable encryption using a new session key and send an LL_ENC_RSP packet in response.
6. Lower Tester sends an LL_START_ENC_REQ packet in response to the master transmissions and receives acknowledgement of the packet.
7. Lower Tester receives an LL_START_ENC_RSP packet encrypted from the IUT and sends a LL_START_ENC_RSP packet encrypted in response.
8. Upper Tester receives an HCI_Encryption_Key_Refresh event.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).

**Expected Outcome**

**Pass Verdict**
The IUT starts the encryption pause procedure upon reception of HCI_LE_Start_Encryption.
The IUT sends the LL_ENC_REQ packet.
The IUT sends the event HCI_Encryption_Key_Refresh_Complete_Event once encryption is resumed.

4.8.6.3 LL/SEC/MAS/BV-03-C [Master Receiving LL_REJECT_IND]

**Test Purpose**
Test that, while executing the Encryption Start Procedure, a master IUT receives an LL_REJECT_IND control packet and notifies it to the Host.

The Lower Tester acts as slave, maintaining a connection, then sends an LL_REJECT_IND packet to the master IUT.

**Reference**
[3] 5.1.3.1

**Initial Condition**
State: Connected Master (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).

**Test Procedure**
Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).

**Figure 4.407: LL/SEC/MAS/BV-03-C [Master Receiving Reject_Ind]**
1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
2. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT and receives an HCI_Command_Status event in response.
3. Lower Tester receives an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector from the IUT.
4. Lower Tester sends an LL_REJECT_IND packet to the IUT with ErrorCode indicating “Unsupported Remote Feature / Unsupported LMP Feature”.
5. Upper Tester receives an HCI_Encryption_Change event from the IUT with error code indicating “Unsupported Remote Feature / Unsupported LMP Feature”.

• Expected Outcome
  
  **Pass Verdict**
  
  Lower Tester receives an LL_ENC_REQ packet, containing the random number ‘ediv’, from the IUT.  
  
  Upper Tester receives an HCI_Encryption_Change event with the ‘Status’ parameter set to ‘Unsupported Remote Feature / Unsupported LMP Feature’ from the IUT.

4.8.6.4 LL/SEC/MAS/BV-04-C [Master Encryption: Sending Data before LL_ENC_RSP]

• Test Purpose
  
  Test, during the encryption start procedure, that a master IUT reports data received after master sends LL_ENC_REQ and before master receives LL_ENC_RSP and does not terminate the established connection.
  
  The Lower Tester acts as in the slave role, first receiving the encryption mode setup request from the IUT, then sends a data packet before sending LL_ENC_RSP.

• Reference
  
  [3] 5.1.3

• Initial Condition
  
  State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).
• Test Procedure

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. Lower Tester receives a LL_ENC_REQ packet including the parameters from step 1.
3. Lower Tester sends a DATA packet with 0xFF as data.
4. Upper Tester receives an HCI_LE_Data_Packet with 0xFF as data.
5. Lower Tester sends an LL_ENC_RSP packet.
6. Lower Tester sends an LL_START_ENC_REQ packet.
7. Lower Tester receives an LL_START_ENC_RSP packet encrypted from the IUT and responds with an LL_START_ENC_RSP packet encrypted.
8. Upper Tester receives an HCI_Encryption_Change event from the IUT, containing the connection handle from the preamble steps’ execution.
9. Lower Tester sends a data packet with encrypted (0x11) as data.
10. Upper Tester receives an HCI_LE_Data_Packet with 0x11 as data.
• **Expected Outcome**

**Pass Verdict**

The IUT does not send a LL_TERMINATE.

The IUT sends an HCI_LE_Data_Packet, with data 0xFF, before sending HCI_Encryption_Change event.

The IUT sends an HCI_Encryption_Change event with status set to “SUCCESS” (0x00)

The IUT sends an HCI_LE_Data_Packet, with data 0x11 after sending HCI_Encryption_Change event.

The IUT does not send an HCI_Disconnection_Complete event

### 4.8.6.5 LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]

• **Test Purpose**

Tests, during the encryption resume procedure, that a master IUT reports data received after master sends LL_PAUSE_ENC_REQ and before master receives LL_PAUSE_ENC_RSP and does not terminate the established connection.

The Lower Tester acts as in the slave role, first accepting the encryption pause request from the IUT, then sends a data packet before sending LL_PAUSE_ENC_RSP.

• **Reference**

[3] 5.1.3.2

• **Initial Condition**

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).
Test Procedure

Figure 4.409: LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. Lower Tester receives an encrypted LL_PAUSE_ENC_REQ packet.
3. Lower Tester sends a data packet with encrypted (0xFF) as data.
4. Upper Tester receives an HCI_LE_Data_Packet with 0xFF as data.
5. Lower Tester sends an encrypted LL_PAUSE_ENC_RSP packet.
6. Lower Tester receives an unencrypted LL_PAUSE_ENC_RSP packet from the IUT. At this point the connection stops being encrypted.
7. Lower Tester receives an LL_ENC_REQ packet to re-enable encryption using a new session key, and sends an LL_ENC_RSP packet in response.
8. Lower Tester sends an LL_START_ENC_REQ packet.
9. Lower Tester receives an LL_START_ENC_RSP packet encrypted from the IUT and responds with an LL_START_ENC_RSP packet encrypted.
10. Upper Tester receives an HCI_Encryption_Key_Refresh event from the IUT.
11. Lower Tester sends a data packet with encrypted (0x11) as data.
12. Upper Tester receives an HCI_LE_Data_Packet with 0x11 as data.

• Expected Outcome

Pass Verdict
The IUT does not send a LL_TERMINATE_IND.
The IUT sends an HCI_LE_Data_Packet, with data 0xFF, before sending HCI_Encryption_Change event.
The IUT sends an HCI_Encryption_Key_Refresh event with status set to “SUCCESS” (0x00)
The IUT sends an HCI_LE_Data_Packet, with data 0x11 after sending HCI_Encryption_Change event.
The IUT does not send an HCI_Disconnection_Complete event.

4.8.6.6 LL/SEC/MAS/BV-06-C [Initiate LE Ping procedure when encryption is enabled]

• Test Purpose
Verify that the IUT as master sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_PING_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.
The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

• Reference
[3] 5.1.8

• Initial Condition
State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)
The connection is encrypted.
The connection is kept idle i.e., no LE-U or LE-C traffic is exchanged.
• Test Procedure

An encrypted connection has been established between the IUT and the Lower Tester

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_PING_RSP.
2. The Lower Tester responds with LL_PING_RSP.

• Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.

• Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.6.7 LL/SEC/MAS/BV-07-C [Responding to LL_PING_REQ]

• Test Purpose

Verify that the IUT as master responds to an LL_PING_REQ sent by the Lower Tester.

• Reference

[3] 5.1.8

• Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)
The connection is encrypted.
The connection is kept idle i.e., no LE-U or LE-C traffic is exchanged.

- **Test Procedure**

![Diagram of LL/SEC/MAS/BV-07-C](image)

**Figure 4.411: LL/SEC/MAS/BV-07-C (Responding to LL_PING_REQ)**

1. The Lower Tester transmits the PDU LL_PING_REQ.
2. The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

- **Expected Outcome**

  **Pass Verdict**

  The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

4.8.6.8 **LL/SEC/MAS/BV-08-C [No response to LL_PING_REQ]**

- **Test Purpose**

  Verify that the IUT as master generates the HCI Authenticated Payload Timeout Expired event when the Lower Tester doesn’t send a packet containing a valid MIC to the IUT within the Authenticated_Payload_Timeout interval.

- **Reference**

  [3] 5.1.8

- **Initial Condition**

  State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

  The connection is encrypted.
• Test Procedure

1. The Upper Tester sets the Authenticated_Payload_Timeout (defined as an IXIT).
2. The Upper Tester unMASKS the HCI Authenticated Payload Timeout Expired event.
3. The Lower Tester sends a data packet containing valid MIC.
4. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.
5. The IUT transmits the PDU LL_PING_REQ to the Lower Tester.
6. The Lower Tester does not respond with LL_PING_RSP.
7. The IUT sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester

    • Expected Outcome

    Pass Verdict

    The IUT transmits the PDU LL_PING_REQ to the Lower Tester and sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester when the Lower Tester doesn't respond with an LL_PING_RSP.

4.8.6.9 LL/SEC/MAS/BV-09-C [Modified Authentication Payload Timeout]

• Test Purpose

Verify that the IUT as master uses the correct value of the Authenticated Payload Timeout (greater than 100 s or less than 5 s) set by the Upper Tester.

• Reference

[3] 5.1.8
- **Initial Condition**
  State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

- **Test Procedure**

  ![Diagram](attachment:image.png)

  **Figure 4.413: LL/SEC/MAS/BV-09-C (Modified Authentication Payload Timeout)**

1. The Upper Tester writes Authenticated_Payload_Timeout (defined as an IXIT) less than 5 s.
2. The Lower Tester sends a data packet containing valid MIC.
3. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged for time greater than Authenticated_Payload_Timeout (defined as an IXIT).
4. The IUT transmits the PDU LL_PING_REQ before time which is less than Authenticated_Payload_Timeout (defined as an IXIT).
5. The Lower Tester responds with LL_PING_RSP. The time between the two packets from the Lower Tester containing a MIC shall not be greater than Authenticated_Payload_Timeout (defined as an IXIT).

- **Expected Outcome**

  **Pass Verdict**
  The IUT transmits the PDU LL_PING_REQ and receives the resulting LL_PING_RSP PDU within the time defined in the IXIT for Authenticated_Payload_Timeout after receiving a packet containing a valid MIC from the Lower Tester.
• Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another Authenticated_Payload_Timeout time.

4.8.6.10 LL/SEC/MAS/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

• Test Purpose

Verify that the IUT as master sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_UNKNOWN_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

• Reference

[3] 5.1.8

• Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

Lower Tester does not support LE Ping Procedure.

The connection is encrypted.

The connection is kept idle i.e., no LE-U or LE-C traffic is exchanged.
• Test Procedure

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after the establishment of an encrypted connection.
2. The Lower Tester responds with LL_UNKNOWN_RSP.
3. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving the LL_UNKNOWN_RSP.
4. The Lower Tester responds with LL_UNKNOWN_RSP.

• Expected Outcome

Pass Verdict
The IUT transmits the LL_PING_REQ PDU to trigger an LL_UNKNOWN_RSP PDU over the air before the 30 s Payload Authentication Timeout expires.

• Notes
The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.6.11 LL/SEC/MAS/BV-11-C [Master Receiving LL_REJECT_EXT_IND]

• Test Purpose
Tests that, while executing the Encryption Start Procedure, a master IUT receives an LL_REJECT_EXT_IND control packet and notifies it to the Host.
The Lower Tester acts as slave, maintaining a connection, then sends an LL_REJECT_EXT_IND packet to the master IUT.

- Reference
  [3] 5.1.3.1

- Initial Condition
  Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

  State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).

![Diagram of test procedure]

Figure 4.415: LL/SEC/MAS/BV-11-C [Master receiving LL_REJECT_EXT_IND]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT and receives an HCI_Command_Status event in response.
2. Lower Tester receives an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector from the IUT. Send an ENC_RSP packet in response.
3. Lower Tester sends an LL_REJECT_EXT_IND PDU to the IUT with RejectOpcode set to “LL_ENC_REQ” and ErrorCode set to ‘0x06’ indicating “PIN or Key missing”.
4. Upper Tester receives an HCI_Encryption_Change event from the IUT with the ‘Status’ parameter set to ‘0x06’ indicating “PIN or key missing”.

- Expected Outcome
  Pass Verdict
  Lower Tester receives an LL_ENC_REQ packet, containing the random number ‘ediv’, from the IUT.
  Upper Tester receives an HCI_Encryption_Change event with the ‘Status’ parameter set to ‘0x06’ indicating “PIN or key missing” from the IUT.
4.8.6.12  LL/SEC/MAS/BV-12-C [Master Start Encryption: Overlapping Procedure]

• Test Purpose
  Tests that an IUT as master can complete the encryption start procedure correctly if an unexpected data channel PDU is received.

• Reference
  [3] 5.1.3.1

• Initial Condition
  Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

• Test Procedure
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and “ir’).
1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. The IUT sends an LL_ENC_REQ packet including the parameters from step 1.
3. The Lower Tester sends a LL_VERSION_IND packet to the IUT followed by the LL_ENC_RSP.
Alternative 1:
1. The IUT should queue the response for the LL_VERSION_IND.
2. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.
3. IUT sends the queued response to LL_VERSION_IND.

Alternative 2:
1. The IUT should ignore the LL_VERSION_IND as the version information has already been exchanged prior to the start of the test procedure.
2. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.

• Expected Outcome

Pass verdict:
Alternative 1:
The IUT sends a queued response to the LL_VERSION_IND packet after the encryption start procedure has completed.

Alternative 2:
The IUT ignores the LL_VERSION_IND from step 3 as the version information has already been exchanged.

Alternatives 1 and 2:
The Upper Tester receives the Encryption_Change_Event from the IUT


• Test Purpose
Tests that an IUT as master can complete the encryption start procedure correctly if an unexpected data channel PDU is received.

• Reference
[3] 5.1.3.1

• Initial Condition
Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

IUT must be capable of processing LL_SLAVE_FEATURE_REQ command

• Test Procedure
Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and “ir”).
1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Receive an HCI_Command_Status event in response.
2. The IUT sends an LL_ENC_REQ packet including the parameters from step 1.
3. The Lower Tester sends a LL_SLAVE_FEATURE_REQ packet to the IUT followed by the LL_ENC_RSP.
4. The IUT should queue the response for the LL_SLAVE_FEATURE_REQ packet.
5. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.
6. IUT sends the queued response to LL_SLAVE_FEATURE_REQ.

- Expected Outcome

**Pass verdict:**
The IUT sends a queued LL_FEATURE_RSP packet after the encryption start procedure has completed.

The Upper Tester receives the Encryption_Change_Event from the IUT.

### 4.8.6.14 LL/SEC/MAS/BV-14-C [Master Receiving unexpected PDU during encryption start]

- **Test Purpose**

Test that a Master IUT which has started the encryption procedure does not respond to an LL_VERSION_IND but instead drops the link.

The Lower Tester acts as a Slave.
• Reference
[3] 5.1.3.1

• Initial Condition
State: Connected Master (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value).

• Test Procedure
Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).

![Diagram of test procedure](image)

*Figure 4.418: LL/SEC/MAS/BV-14-C [Master Receiving unexpected Data Channel PDU during encryption start]*

1. In the preamble steps (connection establishment) use public address type for the Upper Tester and a supported type of address for the Lower Tester.
2. Upper Tester calculates the master portions of the session key diversifier, the initialization vector and a random number.
3. Upper Tester sends an HCI_Start_Encryption command, containing the random number, ‘ediv’ of master portions of the session key diversifier and the initialization vector to the Lower Tester. Receive an HCI_Command_Status_Event in response.
4. The Lower Tester sends LL_ENC_RSP packet, with the slave portions of the session key diversifier and the initialization vector.
5. The ENC_RSP packet is immediately followed by a LL_VERSION_IND packet from the Lower Tester. (In the same connection event if possible).
6. The IUT shall terminate the connection on receiving the LL_VERSION_IND packet.
7. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination from “Connection Terminated Due to MIC Failure (0x3D).

• Expected Outcome
Pass verdict:
The IUT terminates the connection upon receiving the unexpected LL_VERSION_IND packet.

- **Test Purpose**
  Test that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send an encryption response packet.
  
  The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

- **Reference**
  [3] 5.1.3

- **Initial Condition**
  State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- **Test Procedure**
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and ‘ir’).

![Diagram of LL/SEC/MAS/BI-01-C test procedure](image)

*Figure 4.419: LL/SEC/MAS/BI-01-C [Master Encryption Setup: Missing Response]*

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. Lower Tester receives an LL_ENC_REQ packet including the parameters from step 1.
3. Lower Tester acknowledges the encryption request, but does not respond with an LL_ENC_RSP packet.
4. The IUT sends empty data packets until the connection control timer expires.
5. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).
• Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping to maintain the connection.

4.8.6.16  LL/SEC/MAS/Bl-03-C [Master Encryption Setup: Missing Request]

• Test Purpose

Test that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send a start encryption request packet.

The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

• Reference

[3] 5.1.3

• Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout).

• Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘ltk’ and ‘ir’).

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. Lower Tester receives an LL_ENC_REQ packet including the parameters from step 1: with the master portions of the session key diversifier and the initialization vector.
3. Lower Tester acknowledges the encryption request and responds with an LL_ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.
4. Lower Tester does not send LL_START_ENC_REQ packet, but respond to transmissions with slave transmissions using the acknowledgement scheme.

5. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).

- Expected Outcome
  
  **Pass Verdict**
  
  The test procedure executes successfully, with the IUT stopping to maintain the connection.

4.8.6.17 LL/SEC/MAS/BI-04-C [Master Encryption Setup: Missing Acknowledgement]

- Test Purpose
  
  Tests that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send the second start encryption response packet.

  The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

- Reference
  
  [3] 5.1.3

- Initial Condition
  
  State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure
  
  Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘ltk’ and ‘ir’).
1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps’ execution. Receive an HCI_Command_Status event in response.

2. Lower Tester receives an LL_ENC_REQ packet including the parameters from step 1: with the master portions of the session key diversifier and the initialization vector. Lower Tester acknowledges the encryption request and responds with an LL_ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.

3. Lower Tester calculates the session key diversifier and initialization vector. Calculates the diversifier, using the parameters received in step 2. Calculates the long term key from the diversifier using the diversifier hiding key.

4. Lower Tester receives an empty DATA packet from the IUT and responds with an LL_START_ENC_REQ packet.

5. Lower Tester receives an LL_START_ENC_RSP packet encrypted from the IUT. Once the packet is received acknowledges the packet but does not respond with an LL_START_ENC_RSP packet.

6. Lower Tester sends empty data packet. The IUT responds to transmissions with slave transmissions using the acknowledgement scheme until the connection control timer expires.

7. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).

- Expected Outcome

  Pass Verdict

  The test procedure executes successfully, with the IUT stopping to maintain the connection.
4.8.6.18  LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC]

- **Test Purpose**
  Tests that a master IUT terminates a connection upon a MIC failure in a packet received from a slave device.

  The Lower Tester acts in the slave role in an encrypted connection, first transferring data with valid packets, then corrupts the packet contents to cause termination.

- **Reference**
  [3] 3.1, 5.1.3

- **Initial Condition**
  State: Encrypted Master Connection (common identity root, common encryption root, any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- **Test Procedure**
  Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.

![Diagram of the test procedure](image)

**Figure 4.422: LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC]**

1. Lower Tester receives a DATA packet from the IUT and sends a DATA packet in response with 10 bytes of data, flipping a bit in the MIC calculated from the unencrypted packet.
2. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
3. Lower Tester continues slave transmissions.
4. Repeat step 3 until the IUT stops master transmissions up to a time equal to the connection supervision timeout value.

- **Expected Outcome**
  **Pass Verdict**
  The test procedure executes successfully, with the IUT terminating the connection.

  The IUT reports the termination with an HCI event.
4.8.6.19  LL/SEC/MAS/BI-06-C [Master MIC Failure: Corrupt Header]

- Test Purpose
  Test that a master IUT terminates a connection upon a MIC failure in a packet received from a slave device.
  The Lower Tester acts in the slave role in an encrypted connection, first transferring data with valid packets, then corrupts the packet contents to cause termination.

- Reference
  [3] 3.1, 5.1.3

- Initial Condition
  State: Encrypted Master Connection (common identity root, common encryption root, any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure
  Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.

  1. Lower Tester receives a DATA packet from the IUT and transmits a DATA packet in response with 10 bytes of data, flipping a single bit in the header (LLID or RFU) after MIC calculation from the unencrypted packet. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
  2. Lower Tester continues slave transmissions.
  3. Repeat step 2 until the IUT stops master transmissions up to a time equal to the connection supervision timeout value.

*Bluetooth SIG Proprietary*
• Expected Outcome
  
  **Pass Verdict**
  
  The test procedure executes successfully, with the IUT terminating the connection,
  
  The IUT reports the termination with an HCI event.

4.8.6.20  **LL/SEC/MAS/BI-07-C [Master Pause Encryption Sending Data]**

• Test Purpose
  
  Test that a master IUT terminates the established connection when a data packet is received during the encryption pause procedure.

  The Lower Tester acts as in the slave role in an encrypted connection, first accepting the encryption pause request from the IUT, then sends a data packet during the encryption pause procedure to cause termination.

• Reference
  
  [3] 5.1.3

• Initial Condition
  
  State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout).

• Test Procedure

  ![Diagram of Test Procedure](image)

  **Figure 4.424: LL/SEC/MAS/BI-07-C [Master Pause Encryption Sending Data]**
1. As preamble: an encrypted connection is established (IUT acts as master).
2. Upper Tester sends an HCI_LE_Start_Encryption command and receives an 
   HCI_Command_Status in response.
3. Lower Tester receives an encrypted PAUSE_ENC_REQ packet and sends an encrypted 
   PAUSE_ENC_RSP packet in response.
4. Lower Tester receives an unencrypted PAUSE_ENC_RSP packet from the IUT. At this point the 
   connection stops being encrypted.
5. Lower Tester receives an LL_ENC_REQ packet to re-enable encryption using a new session key.
6. Lower Tester sends an LL_ENC_RSP packet.
7. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
8. Upper Tester receives an HCI_Disconnection_Complete event with status set to “Connection 
   Terminated Due to MIC Failure”.

An IUT that conforms to Core Specification v4.0 may send a TERMINATE_IND PDU with error code 
"Connection Terminated Due to MIC Failure (0x3D)” to the Lower Tester between steps 7 and 8.

• Expected Outcome

   Pass Verdict
   The IUT sends an HCI_Disconnection_Complete event with status set to “Connection Terminated 
   Due to MIC Failure”.

4.8.6.21   LL/SEC/MAS/BI-08-C [Master Encryption: Sending Data and Not Response]

• Test Purpose

   Test that a master IUT terminates the established connection when the procedure response timeout 
   timer expires during encryption setup process.

   The Lower Tester acts as in the slave role, first accepting the encryption mode setup request from the 
   IUT, then sends a data packet during the encryption start procedure and then never sending 
   LL_ENC_RSP packet, thus triggering connection termination due to procedure response timeout 
   timer expiring on IUT.

• Reference

   [3] 5.1.3

• Initial Condition

   State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected 
   Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, 
   public own address, common connection interval, common slave latency, common timeout).
• Test Procedure

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an HCI_Command_Status event in response.
2. Lower Tester receives an ENC_REQ packet including the parameters from step 1.
3. Lower Tester acknowledges the encryption request, but does not respond with an ENC_RSP packet.
4. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
5. Upper Tester receives an HCI_Disconnection_Complete event with status set to “LL response timeout”.

• Expected Outcome

Pass Verdict
The IUT sends an HCI_Disconnection_Complete event with status set to “LL response timeout”.

4.8.6.22 LL/SEC/MAS/BI-09-C [Master Encryption: Sending Data and Not Request]

• Test Purpose
Test that a master IUT terminates the established connection when a data packet is received during the encryption start procedure instead of a start encryption request packet.

The Lower Tester acts as in the slave role, first accepting the encryption mode setup request from the IUT, then sends a data packet during the setup sequence.

• Reference
[3] 5.1.3
• **Initial Condition**

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).

• **Test Procedure**

![Diagram](image.png)

**Figure 4.426: LL/SEC/MAS/BI-09-C [Master Encryption: Sending Data and Not Request]**

1. Upper Tester sends an `HCI_LE_Start_Encryption` command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Receive an `HCI_Command_Status` event in response.
2. Lower Tester receives an ENC_REQ packet including the parameters from step 1.
3. Lower Tester acknowledges the encryption request and responds with an ENC_RSP packet.
4. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
5. Upper Tester receives an `HCI_Disconnection_Complete` event with status set to “Connection Terminated Due to MIC Failure”.

An IUT that conforms to Core Specification v4.0 may send a TERMINATE_IND PDU with error code "Connection Terminated Due to MIC Failure (0x3D)" to the Lower Tester between steps 4 and 5.

• **Expected Outcome**

**Pass Verdict**

The IUT sends an `HCI_Disconnection_Complete` event with status set to “Connection Terminated Due to MIC Failure”.

4.9 Data Flow

4.9.1 Both Connected Roles

4.9.1.1 [Transmit Fragmented L2CAP Header]

Test that the IUT correctly transmits packets with fragmented L2CAP headers.

- Test Case IDs
  
  LL/DFL/SLA/BV-01-C
  
  LL/DFL/MAS/BV-01-C

- Reference

  [11] 5.4.2
  
  [12] 7.2.1

- Initial Condition

  State: Connected in the relevant role (any scan interval, any scan window, public peer address, lower tester address, supported type of own address, connection interval, common slave latency, common timeout) to the Lower Tester.
• Test Procedure

1. The Upper Tester sends a L2CAP frame of 28 octets to the IUT with the start fragment containing a Payload length according to Table 4.99 and the rest in one or more continue fragments.

<table>
<thead>
<tr>
<th>Round</th>
<th>Payload Length (octets) (Step b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
2. The Lower Tester receives the L2CAP frame unaltered, possibly unfragmented or fragmented differently.

- **Expected Outcome**
  
  **Pass Verdict**
  
  The Lower Tester receives the L2CAP frame unaltered, irrespective of fragmentation.

### 4.9.1.2 [Receive Fragmented L2CAP Header]

Test that the IUT correctly receives packets with fragmented L2CAP headers.

- **Test Case IDs**
  
  LL/DFL/SLA/BV-02-C
  
  LL/DFL/MAS/BV-02-C

- **Reference**
  
  [11] 5.4.2
  
  [12] 7.2.1

- **Initial Condition**

  State: Connected in the relevant role (any scan interval, any scan window, public peer address, lower tester address, supported type of own address, connection interval, common slave latency, common timeout) to the Lower Tester.
• Test Procedure

![Diagram of test procedure]

**Figure 4.428: [Receive Fragmented L2CAP Header]**

1. The Lower Tester sends an L2CAP frame of 28 octets to the IUT with the start fragment containing a Payload length according to Table 4.100 and the rest in a single continue fragment.

<table>
<thead>
<tr>
<th>Round</th>
<th>Payload Length (octets) (Step b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
2. The Upper Tester receives the L2CAP frame unaltered, possibly unfragmented or fragmented differently.

- Expected Outcome
  - **Pass Verdict**
    The Upper Tester receives the L2CAP frame unaltered, irrespective of fragmentation.

### 4.10 CIS
Tests that the IUT behaves according to the CIS setup and CIS procedures.

#### 4.10.1 Common Variables, Timing, and Procedures

#### 4.10.1.1 Acknowledgement Sequence

Master and slave roles in the CIS tests use variables indicating the state of acknowledgement:

<table>
<thead>
<tr>
<th>When sending</th>
<th>SN has value of</th>
<th>NESN has value of</th>
<th>NPI has value of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data PDU</td>
<td>transmitSeqNum</td>
<td>nextExpectedSeqNum</td>
<td>0b0</td>
</tr>
<tr>
<td>Null PDU</td>
<td>0</td>
<td>nextExpectedSeqNum</td>
<td>0b1</td>
</tr>
</tbody>
</table>

Whether the local variables transmitSeqNum and nextExpectedSeqNum are changed or not changed are determined by the comparison result of the SN/NESN in the Header of the PDU received and the values of current local variables transmitSeqNum and nextExpectedSeqNum.

<table>
<thead>
<tr>
<th>When</th>
<th>Local variable transmitSeqNum</th>
<th>Local variable nextExpectedSeqNum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Flush timeout occurs in the transmitter</td>
<td>transmitSeqNum + 1</td>
<td>No change</td>
</tr>
<tr>
<td>Flush timeout occurs in the receiver</td>
<td>No change</td>
<td>nextExpectedSeqNum + 1</td>
</tr>
<tr>
<td>Received SN = local nextExpectedSeqNum</td>
<td>No change</td>
<td>current nextExpectedSeqNum + 1</td>
</tr>
<tr>
<td>Received SN != local nextExpectedSeqNum</td>
<td>transmitSeqNum + 1, when flush timeout occurs</td>
<td>nextExpectedSeqNum + 1, when flush timeout occurs</td>
</tr>
</tbody>
</table>
### 4.10.1.2 Timing Requirements

The `connSupervisionTimeout` for an ACL with associated CISes shall be greater than twice that of the ISO\_Intervals of the associated CISes.

When verifying the timing requirements, the following requirements must be met:

1. When the Link Layer uses the active clock accuracy during a connected isochronous event to transmit a packet, even if they are in different CISes or in different subevents, or to time intervals between an advertising packet and a packet containing a SCAN\_REQ, AUX\_SCAN\_REQ, CONNECT\_IND or AUX\_CONNECT\_REQ PDU, the instantaneous timings shall not deviate more than 2 μs from the average timing. That is, the start of a packet shall be transmitted 150±2 μs after the end of the previous packet.

2. When the Link layer uses the sleep clock accuracy to transmit, the instantaneous timing of the anchor point shall not deviate more than 16 μs from the average timing. That is, the tolerance allowed on either side of the anchor point is sleep clock accuracy plus 16 μs.

3. The clock used to time packet reception may have any accuracy, but the receiving device will need to allow for this, and adjust the time when it starts and continues listening for packets accordingly.

### 4.10.1.3 Default Values for Set CIG Parameters Commands

When using either the HCI\_LE\_Set\_CIG\_Parameters or HCI\_LE\_Set\_CIG\_Parameters\_Test commands, the following table defines common default parameters for this section. The test case may specify different values.

<table>
<thead>
<tr>
<th>When</th>
<th>Local variable transmitSeqNum</th>
<th>Local variable nextExpectedSeqNum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received NESN = local transmitSeqNum</td>
<td>transmitSeqNum + 1, when flush timeout occurs. Need to re-transmit last sent PDU.</td>
<td>Need to retransmit last sent PDU except when the flush timeout is reached, then nextExpectedSeqNum = nextExpectedSeqNum + 1.</td>
</tr>
<tr>
<td>Received NESN != local transmitSeqNum</td>
<td>current transmitSeqNum + 1</td>
<td>No change</td>
</tr>
<tr>
<td>Received SN = SN in last received PDU</td>
<td>No change. PDU is ignored.</td>
<td></td>
</tr>
<tr>
<td>Invalid CRC match</td>
<td>transmitSeqNum + 1, when flush timeout occurs</td>
<td>No change except when flush timeout is reached, then nextExpectedSeqNum = nextExpectedSeqNum + 1.</td>
</tr>
<tr>
<td>Set CIG Parameters</td>
<td>Set CIG Parameters Test</td>
<td>Default Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>SDU_Interval_M_To_S</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SDU_Interval_S_To_M</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ISO_Interval</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>CIS_Count</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Slaves_Clock_Accuracy</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Packing</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Framing</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NSE</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Max_SDU_M_To_S</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Max_SDU_S_To_M</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Max_PDU_M_To_S</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Max_PDU_S_To_M</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PHY_M_To_S</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PHY_S_To_M</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FT_M_To_S</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>FT_S_To_M</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>BN_M_To_S</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>BN_S_To_M</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Transport_Latency_M_To_S</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Transport_Latency_S_To_M</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>RTN_M_To_S</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>RTN_S_To_M</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
Note 1: Set to the maximum SDU size as defined by the bandwidth requirements specified in [17] Section 2.1 for unframed PDUs and Section 2.2 for framed PDUs, or MAX SDU Length defined in IXIT, whichever is smaller.

4.10.1.4 Configuring the ISO Data Path for HCI
The HCI_LE_Setup_ISO_Data_Path shall always be configured to use the HCI for input and output. Input_Data_Path shall be set to 0x00 (HCI transport), and likewise, Output_Data_Path shall be set 0x00 (HCI transport).

4.10.1.5 Identification of CISes
When the notation CIS(n) is used to refer to a CIS within its CIG, n refers to the order of the CIS within its CIG.

When creating CISes, the Upper Tester shall assign the CIS_ID of each CIS in the order it is created, starting with 1. When referring to a CIS by its creation order, it can be referred to using the notation “CIS with CIS_ID = i”, where i refers to the creation order. For example, the CIS with CIS_ID = 1 is the first CIS created, while the CIS with CIS_ID = 3 is the third CIS created.

4.10.2 MAS
Tests that the IUT behaves according to the Connected Isochronous Stream master creation procedures.

4.10.2.1 [CIS Setup Procedure, Master Initiated]

- Test Purpose
  Verify that a master IUT can set up a Connected Isochronous Stream and maintain a connection with a slave device while sending data packets in the Connected Isochronous Stream.

- Reference
  [14] Section 5.1.15

- Initial Condition
  The Isochronous Channels (Host Support) FeatureSet bit is set.

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle on the PHY specified in Table 4.101: Master Test Case Direction Specific Configurations.

The Lower Tester acts in the slave role.

TSPX_conn_interval is the Connection Interval as specified in the IXIT [9] entry.
Test Procedure

1. The Upper Tester sends an HCI_LE_Set_CIG_Parameters_Test command to the IUT with CIS_Count set to 1, BN, FT, NSE, PHY_M_To_S[], PHY_S_To_M[] and ISO_Interval to be set to the values specified in Table 4.101 and Table 4.102. Any remaining values are assigned the default values as specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands. The Upper Tester receives a successful HCI_Command_Complete event with a valid Connection_Handle from the IUT and CIS_Count = 1.
2. The Upper Tester sends an HCI_LE_Create_CIS command to the IUT with the
ACL_Connection_Handle of the established ACL connection and CIS_Count set to 1. The Upper
Tester receives a Status of Success from the IUT.
3. The Lower Tester receives an LL_CIS_REQ PDU from the IUT with all fields set to valid values.
   CIS_Offset_Min is a value between 500µs and TSPX_conn_interval, CIS_Offset_Max is a value
   between CIS_Offset_Min and the CIS_Offset_Max value as calculated in [14] Section 2.4.2.29
   using TSPX_conn_interval as the value of connInterval, and connEventCount is the reference
   event anchor point for which the offsets applied.
4. The Lower Tester sends an LL_CIS_RSP PDU to the IUT.
5. The Lower Tester receives an LL_CIS_IND from the IUT where the CIS_Offset is the time (ms)
   from the start of the ACL connection event in connEvent Count to the first CIS anchor point, the
   CIS_Sync_Delay is CIG_Sync_Delay minus the offset from the CIG reference point to the CIS
   anchor point in µs, and the connEventCount is the CIS_Offset reference point.
6. The IUT sends a CIS Null PDU to the Lower Tester and the Lower Tester responds with a CIS
   Null PDU. Alternately, the IUT sends an empty Data PDU, which the Lower Tester acknowledges.
   These exchanges will continue until data is exchanged between the IUT and the Lower Tester
   in later steps.
7. The Upper Tester receives a successful HCI_LE_CIS_Established event with the NSE, BN, FT,
   and Max_PDU parameters as set in step 1 from the IUT, after the first CIS packet sent by the LT.
   The Connection_Handle parameter is set to the value provided in the HCI_LE_Create_CIS
   command.
8. The Upper Tester orders the IUT to send data packets to the Lower Tester.
9. The Lower Tester receives CIS data PDUs from the IUT in each sub-event of the CIS and
   acknowledges those PDUs.
10. Repeat step 9 for 50 ÷ BN isochronous events starting with the first event where a CIS data PDU
    with non-zero payload is received.

- Expected Outcome

<table>
<thead>
<tr>
<th>Test Case</th>
<th>M_To_S</th>
<th>Payload (PDU)</th>
<th>S_To_M</th>
<th>Payload (PDU)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PHY</td>
<td>BN</td>
<td>FT</td>
<td>PHY</td>
</tr>
<tr>
<td>4.10.2.1.1 LL/CIS/MAS/BV-01-C</td>
<td>LE 1M PHY</td>
<td>2 1 130</td>
<td>LE 1M PHY</td>
<td>2 1 130</td>
</tr>
<tr>
<td>4.10.2.1.2 LL/CIS/MAS/BV-02-C</td>
<td>LE 2M PHY</td>
<td>2 2 251</td>
<td>LE 2M PHY</td>
<td>2 1 251</td>
</tr>
<tr>
<td>4.10.2.1.3 LL/CIS/MAS/BV-25-C</td>
<td>LE Coded PHY</td>
<td>1 1 50</td>
<td>LE Coded PHY</td>
<td>1 1 50</td>
</tr>
<tr>
<td>4.10.2.1.4 LL/CIS/MAS/BV-31-C</td>
<td>LE 2M PHY</td>
<td>2 1 130</td>
<td>LE 1M PHY</td>
<td>2 1 130</td>
</tr>
<tr>
<td>4.10.2.1.5 LL/CIS/MAS/BV-32-C</td>
<td>LE 1M PHY</td>
<td>1 1 50</td>
<td>LE Coded PHY, S=8</td>
<td>0 1 0</td>
</tr>
</tbody>
</table>

Table 4.101: Master Test Case Direction Specific Configurations
### Test Case Additional Configurations

<table>
<thead>
<tr>
<th>Test Case</th>
<th>NSE</th>
<th>SDU_Interval_M_to_S</th>
<th>SDU_Interval_S_To_M</th>
<th>ISO_Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CIS/MAS/BV-01-C</td>
<td>4</td>
<td>10 ms (0x2710)</td>
<td>10 ms (0x2710)</td>
<td>20 ms (0x10)</td>
</tr>
<tr>
<td>LL/CIS/MAS/BV-02-C</td>
<td>4</td>
<td>20 ms (0x4E20)</td>
<td>10 ms (0x2710)</td>
<td>20 ms (0x10)</td>
</tr>
<tr>
<td>LL/CIS/MAS/BV-25-C</td>
<td>2</td>
<td>40 ms (0x9C40)</td>
<td>40 ms (0x9C40)</td>
<td>40 ms (0x20)</td>
</tr>
<tr>
<td>LL/CIS/MAS/BV-31-C</td>
<td>4</td>
<td>10 ms (0x2710)</td>
<td>10 ms (0x2710)</td>
<td>20 ms (0x10)</td>
</tr>
<tr>
<td>LL/CIS/MAS/BV-32-C</td>
<td>2</td>
<td>40 ms (0x9C40)</td>
<td>40 ms (0x9C40)</td>
<td>40 ms (0x20)</td>
</tr>
</tbody>
</table>

#### Table 4.102: Master Test Case Additional Configurations

**Pass Verdict**

- In step 1, the Upper Tester receives a successful HCI\_Command\_Complete event with a valid Connection\_Handle from the IUT and CIS\_Count = 1.
- In step 3, the Lower Tester receives an LL\_CIS\_REQ PDU from the IUT with all the fields set to valid values, including valid values for CIS\_Offset\_Min and CIS\_Offset\_Max as specified.
- In step 7, the Upper Tester receives a successful HCI\_LE\_CIS\_Established. The Connection\_Handle parameter is set to the value provided in the HCI\_LE\_Create\_CIS command.
- In step 10, the Lower Tester receives at least 48 of the CIS data PDUs sent by the IUT in the CIS. The length of the data sent by the IUT matches that specified in Table 4.102.

### 4.10.2.2 LL/CIS/MAS/BV-03-C [CIS Setup Procedure, Master Initiated, Unsupported]

- **Test Purpose**
  Tests that a master IUT can initiate the Connected Isochronous Stream setup procedure and correctly handles the case where the slave does not support the feature.

- **Reference**
  Section [14] 5.1.15

- **Initial Condition**
  The Isochronous Channels (Host Support) FeatureSet bit is set.

  An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

  The Lower Tester acts in the slave role.
Test Procedure

1. The Upper Tester sends an HCI_LE_Set_CIG_Parameters command to the IUT with valid parameters and receives a successful HCI_Command_Complete event.
2. The Upper Tester sends an HCI_LE_Create_CIS command with the ACL_Connection_Handle of the established ACL and valid Connection_Handle from the IUT received in step 1.
3. The Lower Tester receives an LL_CIS_REQ PDU from the IUT with all fields set to valid values.
4. The Lower Tester sends an LL_REJECT_EXT_IND to the IUT with a valid reason code.
5. The Upper Tester receives an HCI_LE_CIS_Established event from the IUT with a status failure. The Status field has the same value as the LL_REJECT_EXT_IND PDU in step 4. Followed by the Upper Tester terminating the ACL connection.

Expected Outcome

Pass Verdict
- In step 1, the IUT sends a successful HCI_Command_Complete event.
- In step 3, the IUT sends an LL_CIS_REQ to the Lower Tester with valid values.
- In step 5, the IUT send an HCI_LE_CIS_Established event with a status failure that is the same value from the LL_REJECT_EXT_IND PDU that the IUT received from the Lower Tester.
- The ACL connection is disconnected.

Inconclusive Verdict
- If the IUT autonomously initiates a feature exchange before step 3, the test ends with an Inconclusive Verdict.
### 4.10.2.3 LL/CIS/MAS/BV-04-C [New Channel Map]

- **Test Purpose**
  Tests that the IUT updates the channel map for a CIS when the channel map for the connection is updated.

- **Reference**
  [14] Sections 5.1.2, 2.4.2.2

- **Initial Condition**
  A CIS has been established.
  The initial channel map is 0x1FFFFFFFFF (i.e., all channels enabled).
  The Lower Tester acts in the slave role.

- **Test Procedure**

  ![Diagram](image)

  **Figure 4.432: LL/CIS/MAS/BV-04-C [New Channel Map]**

  1. The Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command with the Channel_Map parameter set to 0x1249249249 to the IUT and receives an HCI_Command_Complete event in response.
  2. The Lower Tester receives an LL_CHANNEL_MAP_IND command with the ChM field in CtrData field value 0x1249249249, and the instant field is equal to connEventCount.
  3. The Upper Tester orders the IUT to send data packets to the Lower Tester.
  4. The Lower Tester acknowledges the data packets sent by the IUT in the CIS.
  5. Repeat steps 3 and 4 50 times.

- **Expected Outcome**

  **Pass Verdict**
  - In step 1, the IUT sends an HCI_Command_Complete event with success status.
  - In step 2, the IUT sends an LL_CHANNEL_MAP_IND command with the ChM field in CtrData field value 0x1249249249, and a 2-octet value in the instant field.
  - In step 4, the Lower Tester acknowledges at least 48 of the data packets from the IUT.
4.10.2.4 LL/CIS/MAS/BV-10-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Interleaved CIG]

- **Test Purpose**
  Test that a master IUT can send and receive data on multiple CISes in an interleaved arrangement on a single CIG across multiple connections. The Sub_Interval, Sync_Delay, and Anchor point for each CIS on each connection are verified to refer to the same instant in time.

- **Reference**
  [14] Section 4.5.14.2

- **Initial Condition**
  Connected Isochronous Streams have been established between the IUT and Lower Tester 1 and Lower Tester 2 according to the following initial state and in the same CIG:

  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.103)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x01 (Interleaved)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04, 0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251, 251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>default, default</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251, 251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>default, default</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01, 0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01, 0x01</td>
</tr>
</tbody>
</table>
### Table 4.103: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bn_m2s[]</td>
<td>0x03, 0x03</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x03, 0x03</td>
</tr>
</tbody>
</table>

Two Lower Testers (or instances) are set up acting as the two slave devices.

The Lower Testers act in the slave role.

- **Test Procedure**

  1. The Upper Tester orders the IUT to send a payload with size 251 bytes to the Lower Testers.
  2. Lower Tester 1 receives the payload PDU in the first subevent on CIS(1).
  3. Lower Tester 1 sends an Ack T_IFS after receiving the payload PDU.
  4. Lower Tester 2 receives the payload PDU in the first subevent on CIS(2).
  5. Lower Tester 2 sends an Ack T_IFS after receiving the payload PDU.
  6. When CIS(1) subevent interval ends, repeat steps 3–5 in the next subevent.
  7. When CIS(2) subevent interval ends, repeat steps 6 and 7 in the next subevent.
  8. The Lower Tester observes that the time CIS_Sync_Delay minus the offset from the CIG reference point to the CIS(1) anchor point in μs is the observed CIS_Sync_Delay(a). The time between the ACL connection event for the first received CIS and the start of the first subevent on the same CIS is the observed CIS_Offset(a). The Lower Tester validates that CIS_Offset(a) = CIS_Offset in the LL_CIS_IND associated with CIS sent during setup of the CISes.

---

*Figure 4.433: LL/CIS/MAS/BV-10-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Interleaved CIG]*
9. The Lower Tester observes that the time CIG.Sync_Delay minus the offset from the CIG reference point to the CIS(2) anchor point in $\mu$s is the observed CIS.Sync_Delay(b). The time between the ACL connection event for the second received CIS and the start of the first subevent on the same CIS is the observed CIS_Offset(b). The Lower Tester validates that the CIS_Offset(b) = CIS_Offset in the LL_CIS_IND associated with CIS sent during setup of the CISes.

10. When CIS(1) interval ends, repeat steps 1–3 for CIS(1) in the next isochronous event.

11. When CIS(2) interval ends, repeat steps 4 and 5 for CIS(2) in the next isochronous event.

12. When CIS(1) subevent interval ends, repeat steps 1–3 in the next subevent.

13. When CIS(2) subevent interval ends, repeat steps 4 and 5 in the next subevent.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends the payload PDU at CIS(1) Offset from ACL1.
- In step 4, the IUT sends the payload PDU at CIS(2) Offset from ACL2.
- In steps 10 and 11, the results repeat.
- In steps 12–13, the results repeat.

4.10.2.5 LL/CIS/MAS/BV-11-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Sequential]

• Test Purpose

Test that a master IUT can send and receive data in multiple CISes in a sequential arrangement on a single CIG across multiple connections. The Sub_Interv, Synchronization_Delay, and Anchor point for each CIS on each connection are verified to refer to the same instant in time.

• Reference

[14] Section 4.5.14.2

• Initial Condition

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

The Lower Testers 1 and 2 both act in the slave role representing slave 1 and 2, respectively, observes the data in the packets from the master IUT, and sends response back. The Upper Tester submits data for the master IUT to transmit and observes the timing of the CIS subevents.

Each CIS is set according to the following Initial State:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.104)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table 4.104: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>2</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x02</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>
Figure 4.434: LL/CIS/MAS/BV-11-C [Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Sequential]

1. The Upper Tester orders the IUT to send a payload of size 251 to the Lower Tester.
2. Lower Tester 1 receives the payload PDU in the first subevent on CIS(1).
3. Lower Tester 1 sends an Ack, T_IFS after receiving the payload PDU.
4. Repeat steps 1–3 for Lower Tester 2 using CIS(2).
5. The Lower Tester observes that the time CIG_Sync_Delay minus the offset from the CIG reference point to the CIS(1) anchor point in µs is the observed CIS_Sync_Delay(a). The time between the ACL connection event for the first received CIS and the start of the first subevent on the same CIS is the observed CIS_Offset(a). The Lower Tester validates that the CIS_Offset(a) = CIS_Offset in the LL_CIS_IND in the associated CIS.
6. The Lower Tester observes that the time CIG_Sync_Delay minus the offset from the CIG reference point to the CIS(2) anchor point in µs is the observed CIS_Sync_Delay(b). The time between the ACL connection event for the second received CIS and the start of the first subevent on the same CIS is the observed CIS_Offset(b). The Lower Tester validates that the CIS_Offset(b) = CIS_Offset in the LL_CIS_IND in the associated CIS.
7. When CIS(1) interval ends, repeat steps 1–6 without using CIS(1) Offset from ACL1, but right after the first CIS(1) interval ends.

- **Expected Outcome**
  - **Pass Verdict**
    - In step 2, the IUT sends the payload PDU at CIS(1) Offset from ACL1.
    - In step 4, the IUT sends the second payload PDU when CIS(1) subevent 1 CIS interval 1 ends or when CIS(1)’s second subevent interval starts.
    - In step 5, Lower Tester verifies that CIS_Offset(a) = CIS_Offset in the LL_CIS_IND associated with CIS sent in step 1 or step 2.
- In step 6, Lower Tester verifies that CIS_Offset(b) = CIS_Offset in the LL_CIS_IND associated with CIS sent in step 1 or step 2.

4.10.2.6 LL/CIS/MAS/BV-20-C [Set Encryption After CIS Established]

- **Test Purpose**
  Tests that the IUT returns an error if the HCI_LE_Start_Encryption command is sent after a CIS has been established.

- **Reference**
  [14] Vol. 6, Part B, Section 5.1.3.1  
  [16] Vol. 6, Part D, Section 6.6

- **Initial Condition**
  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.105)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
</tbody>
</table>
### Test Procedure

**Table 4.105: State Variable Values**

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

1. The Upper Tester sends an `HCI_LE_Start_Encryption` command with valid parameters to the IUT.
2. The Upper Tester receives the error code Command Disallowed (0x0C).

#### Expected Outcome

**Pass Verdict**

- In step 2, the IUT sends an error code Command Disallowed (0x0C).

### 4.10.2.7 Connected Isochronous Stream Using Non-Test Command, Master Initiated

#### Test Purpose

Verify that a master IUT can set up a Connected Isochronous Stream using the LE Setup CIG Parameters Command and maintain a connection with a slave device while sending data packets in the Connected Isochronous Stream.

#### Reference

[15] Section 7.8.97

#### Initial Condition

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

The IUT uses the PHY as specified in Table 4.106: Master Test Case Configurations.

The Lower Tester acts in the slave role.

Max Supported CIGs is specified in the IXIT values.
### Test Procedure

1. The Upper Tester sends an `HCI_LE_Set_CIG_Parameters` command to the IUT with default parameters but with framing enabled, and receives a success response from the IUT and `CIS_Count = 1`.

2. The Upper Tester sends an `HCI_LE_Create_CIS` command to create a single CIS and receives a success response from the IUT.

3. The Lower Tester receives an `LL_CIS_REQ PDU` from the IUT with all fields set to valid values. `CIS_Offset_Min` is a value between 500µs and TSPX_conn_interval, `CIS_Offset_Max` is a value between `CIS_Offset_Min` and the `CIS_Offset_Max` value as calculated in [14] Section 2.4.2.29 using TSPX_conn_interval as the value of connInterval, and `connEventCount` is the reference event anchor point for which the offsets applied.

4. The Lower Tester sends an `LL_CIS_RSP PDU` to the IUT.

5. The Lower Tester receives an `LL_CIS_IND` from the IUT where the `CIS_Offset` is the time (ms) from the start of the ACL connection event in `connEventCount` to the first CIS anchor point, the `CIS_Sync_Delay` is `CIG_Sync_Delay` minus the offset from the CIG reference point to the CIS anchor point in µs, and the `connEventCount` is the `CIS_Offset` reference point.

---

**Figure 4.436: Connected Isochronous Stream Using Non-Test Command, Master Initiated**

- **Lower Tester**
- **IUT**
- **Upper Tester**

ACL Connection Established. IUT is Master.

- `HCI_LE_Create_CIS (CIS_Count=1)`
  - `HCI_Command_Status` (Status: 0x00)
  - `HCI_LE_Setup_ISO_Data_Path`
    - `HCI_Command_Complete` (Status: 0x00)
  - `HCI_LE_Set_CIG_Parameters`
    - (Valid Default Parameters, Framed)
    - `HCI_Command_Status` (Status: 0x00, CIS_Count=1)
  - `HCI_LE_CIS_Established event` (Connection_Handle)
    - `HCI_LE_Setup_ISO_Data_Path`
      - `HCI_Command_Complete` (Status: 0x00)
    - `HCI_ISO_Data_Packets`
      - ISO Data Packets
        - (Framed)
  - `LL_CIS_REQ PDU`
    - (CIS_Offset_Min, CIS_Offset_Max)
      - `LL_CIS_IND`
        - ISO Data Packet
          - (empty)
    - `LL_ACK`
      - ISO Data Packets
        - (Framed)
  - `HCI_LE_CIS_Established event` (Connection_Handle)
6. The Upper Tester receives an HCI_LE_CIS_Established event indicating success, after the first CIS packet sent by the LT. The Connection_Handle parameter is set to the value provided in the HCI_LE_Create_CIS command.

7. The Upper Tester sends an HCI_LE_Setup_ISO_Data_Path command and receives a success response from the IUT.

8. The Upper Tester sends HCI ISO data packets over the CIS and the Lower Tester receives framed ISO data.

• **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.10.2.7.1 LL/CIS/MAS/BV-26-C</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>4.10.2.7.2 LL/CIS/MAS/BV-27-C</td>
<td>LE 2M PHY</td>
</tr>
<tr>
<td>4.10.2.7.3 LL/CIS/MAS/BV-28-C</td>
<td>LE Coded PHY</td>
</tr>
</tbody>
</table>

*Table 4.106: Master Test Case Configurations*

**Pass Verdict**

- In step 1, the Upper Tester receives a success response from the IUT and CIS_Count = 1.
- In step 2, the Upper Tester receives a success response from the IUT.
- In step 3, the Lower Tester receives an LL_CIS_REQ PDU from the IUT with all fields set to valid values, including valid values for CIS_Offset_Min and CIS_Offset_Max as specified.
- In step 5, the Lower Tester receives an LL_CIS_IND from the IUT as described.
- In step 6, the Upper Tester receives an HCI_LE_CIS_Established event indicating success. The Connection_Handle parameter is set to the value provided in the HCI_LE_Create_CIS command.
- In step 7, the Upper Tester receives a success response from the IUT.
- In step 8, the Lower Tester receives framed ISO data.
- The calculated transport latencies of all the CISes as defined in [17] Section 3.2.1 is less than or equal to the maximum transport latencies specified in the HCI_LE_Set_CIG_Parameters command.

**4.10.2.8 LL/CIS/MAS/BV-30-C [Isochronous Channels Host Support Feature Bit]**

• **Test Purpose**
Tests that a master IUT only creates a CIS when the Isochronous Channels (Host Support) feature bit is set.

• **Reference**
Section [14] 5.1.15

• **Initial Condition**
The Isochronous Channels (Host Support) feature bit is in its power on reset value and has not been set or cleared.

The Lower Tester acts in the slave role.
• Test Procedure

For each round 1 to 3
   If feature enable set this round
       HCI_LE_Set_Host_Feature
          (Bit_Number, Bit_Value)
       HCI_Command_Complete
          (Status)
   If CIS successful this round
       ACL Connection Established. IUT is Master.

   For each round 1 to 3
       If feature enable set this round
           HCI_LE_Set_Host_Feature
              (Bit_Number, Bit_Value)
           HCI_Command_Complete
              (Status)
       ACL Connection Established. IUT is Master.

   ACL Connection Established. IUT is Master.
       HCI_LE_Set_CIG_Parameters
          (CIG_ID)
       HCI_Command_Complete
          (Status: 0x00)
       HCI_LE_Create_CIS
          (ACL_Connection_Handle, Connection_Handle)
       HCI_Command_Status
          (Status)

   LL_CIS_REQ
   LL_CIS_RSP
   LL_CIS_IND
   ACL Connection Disconnected.

Figure 4.437: LL/CIS/MAS/BV-30-C [Isochronous Channels Host Support Feature Bit]

1. The Upper Tester sends an HCI_LE_Set_Host_Feature with Bit_Number set to 30 (not a host
controlled feature bit) and Bit_Value set to 0b1. The Upper Tester receives an
   HCI_Command_Complete response with an error code Unsupported Feature or Parameter Value
   (0x11).
2. The IUT establishes an ACL connection with the Lower Tester as Slave.
3. The Upper Tester sends an HCI_LE_Set_Host_Feature with Bit_Number set to 32 (Isochronous
   Channels feature bit) and Bit_Value set to 0b1. The Upper Tester receives an
   HCI_Command_Complete response with an error code Command Disallowed (0x0C).
4. The IUT disconnects the ACL connection from the Lower Tester.
5. Repeat steps 6–12 for each round specified in Table 4.107.
6. If the table indicates “Set Feature Enable” is true for this round, then the Upper Tester sends an
   HCI_LE_Set_Host_Feature command to the IUT with the Bit_Number set to 32 (Isochronous
Channels) and the Bit_Value set to the value specified in the table. The Upper Tester receives an HCI_Command_Complete event from the IUT.

7. The IUT establishes an ACL connection with the Lower Tester as Slave.

8. The Upper Tester sends an HCI_LE_Set_CIG_Parameters command to the IUT using the default parameters specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands and receives a successful HCI_Command_Complete event.

9. The Upper Tester sends an HCI_LE_Create_CIS command with the ACL_Connection_Handle of the established ACL and valid Connection_Handle.

Table indicates a CIS is created this round

10. The Upper Tester receives a successful HCI_Command_Status event.

11. The IUT establishes a CIS with the Lower Tester.

Table indicates a CIS is not created this round

10. The Upper Tester receives an HCI_Command_Status event with the status set to error code Command Disallowed (0x0C).

12. The IUT disconnects the ACL connection from the Lower Tester.

<table>
<thead>
<tr>
<th>Round</th>
<th>Set Feature Enable</th>
<th>Bit_Value</th>
<th>CIS Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>False</td>
<td>n/a</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>True</td>
<td>0b1</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>True</td>
<td>0b0</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 4.107: Num_BIS Values

- Expected Outcome

  Pass Verdict
  - In step 1, the Upper Tester receives an HCI_Command_Complete response with an error code Unsupported Feature or Parameter Value (0x11).
  - In step 3, the Upper Tester receives an HCI_Command_Complete response with an error code Command Disallowed (0x0C).
  - In rounds 2 and 3 and in step 6, the IUT accepts the HCI_LE_Set_Host_Feature command.
  - In rounds 1 and 3, the IUT returns error code Command Disallowed (0x0C) when the Upper Tester attempts to create a CIS.
  - In round 2, the IUT successfully creates a CIS with the Lower Tester.
  - When the IUT disconnects the ACL connection from the Lower Tester, no LL_CIS_TERMINATE_IND PDUs are generated by the IUT.

4.10.2.9 LL/CIS/MAS/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG, Master]

- Test Purpose

  Test that the master IUT can send and receive data on multiple CISes in an interleaved arrangement on a single CIG on a single connection.
• Reference

[14] Section 4.5.14.2

• Initial Condition

Connected in the master role as defined in the following initial state:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.108)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x01 (Interleaved)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>2</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

Table 4.108: State Variable Values
Test Procedure

Figure 4.438: LL/CIS/MAS/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG, Master]
1. The Upper Tester Sends 4 HCI ISO Data Packets to the IUT, two for each CIS.
2. The IUT sends the first CIS(1) payload PDU to the Lower Tester on CIS(1). Note the CIS_ID of CIS(1).
3. The Lower Tester sends an Ack after receiving the payload PDU from the IUT on CIS(1).
4. The IUT sends the first CIS(2) payload PDU to the Lower Tester on CIS(2). Note the CIS_ID of CIS(2).
5. The Lower Tester sends an Ack after receiving the payload PDU from the IUT on CIS(1).
6. The IUT sends the second CIS(1) payload PDU to the Lower Tester on CIS(1).
7. The Lower Tester sends its first CIS(1) payload PDU to the IUT on CIS(1) after receiving the payload PDU.
8. The IUT receives the payload PDU 1 from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
9. The IUT sends the second CIS(2) payload to the Lower Tester on CIS(2).
10. The Lower Tester sends its first CIS(2) payload to the IUT on CIS(2).
11. The IUT receives the payload PDU 1 from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
12. The IUT sends an Ack to the Lower Tester on CIS(1).
13. The Lower Tester sends its second CIS(1) payload to the IUT on CIS(1).
14. The IUT receives the payload PDU 2 from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
15. The IUT sends an Ack to the Lower Tester on CIS(2).
16. The Lower Tester sends its second CIS(2) payload to the IUT on CIS(2).
17. The IUT receives the payload PDU 2 from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
18. The IUT sends an Ack to the Lower Tester on CIS(1).
19. The IUT sends an Ack to the Lower Tester on CIS(2).
20. Repeat all prior steps three times.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends a CIS payload PDU to the Lower Tester on CIS(1). The payload is one of the first payloads provided by the Upper Tester for a CIS to the IUT.
- In step 4, the IUT sends a CIS payload PDU to the Lower Tester on CIS(2). The payload is the other first payload provided by the Upper Tester for a CIS to the IUT.
- In step 6, the IUT sends a second CIS payload PDU to the Lower Tester on CIS(1). The payload is the second payload provided by the Upper Tester for the CIS with the CIS_ID in step 2.
- In step 8, the IUT receives the payload PDU from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
- In step 9, the IUT sends a second CIS payload to the Lower Tester on CIS(2). The payload is the second payload provided by the Upper Tester for the CIS with the CIS_ID in step 4.
- In step 11, the IUT receives the payload PDU from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
- In step 12, the IUT sends an Ack to the Lower Tester on CIS(1).
- In step 14, the IUT receives the payload PDU from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
- In step 15, the IUT sends an Ack to the Lower Tester on CIS(2).
- In step 17, the IUT receives the payload PDU from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
- In step 18, the IUT sends an Ack to the Lower Tester on CIS(1).
- In step 19, the IUT sends an Ack to the Lower Tester on CIS(2).
- The prior results in steps 1–19 repeat three times.

**4.10.2.10 LL/CIS/MAS/BV-09-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential, Master]**

- Test Purpose
  Test that the master IUT can send and receive data on multiple CISes in a sequential arrangement in a single Connected Isochronous Group on a single connection.

- Reference
  [14] Section 4.5.14.2

- Initial Condition
  Connected in the master role as defined in the following initial state:

  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.109)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x0C350 (50 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>2</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
</tbody>
</table>
### Test Procedure

1. The Upper Tester orders the IUT to send two payload PDUs to the Lower Tester to be sent on CIS(1).
2. The Lower Tester receives a payload PDU on CIS(1).
3. The Lower Tester sends an Ack and a payload PDU T_IFS after receiving the payload PDU from the IUT.
4. The IUT sends an Ack and the second payload PDU, after receiving the payload PDU from the Lower Tester on CIS(1).
5. The Lower Tester sends an Ack T_IFS after receiving a second payload PDU from the IUT.
6. When CIS(1) interval ends, repeat steps 1–5 on CIS(2).
7. Repeat steps 1–6.

- Expected Outcome

  Pass Verdict
  - In step 2, the IUT sends a payload PDU at CIS(1).
  - In step 3, the IUT receives and Ack the payload PDU at CIS(1).
  - In step 4, the IUT sends a second payload PDU at CIS(1)’s subevent 2.
  - In step 6, the IUT sends the payload PDU on CIS(2) and sends the second payload PDU at CIS(2)’s subevent 2.
  - In steps 1–6, the results repeat.

4.10.2.11 [Acknowledgement Scheme, Master]

- Test Purpose
  Test that the master IUT can use the acknowledgement scheme.

- Reference
  [14] Section 4.5.9

- Initial Condition
  Connected in the master role.

  Note: "default" refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.110)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>0x02</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>0x02</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
</tbody>
</table>
### Test Procedure

1. The Upper Tester orders the IUT to send four separate payload PDUs to the Lower Tester.
2. The Lower Tester receives the payload PDU, and in the PDU header the SN field and the NESN field match and the NESN to match the next SN. The CIE and NPI fields are set to 0b0.
3. The Lower Tester sends a CIS data PDU with empty payload and Ack T_IFS after receiving the payload PDU from the IUT. The Lower Tester observes the acknowledgement scheme by using the next SN for each payload PDU sent where the NESN in the packet received is the next SN and by using the next NESN where the current NESN matches the SN in the PDU correctly received.

4. Repeat steps 2–3 for the second and third payload PDUs, except the Lower Tester acknowledges the payload but does not send a CIS data PDU with empty payload.

5. The Lower Tester receives the fourth payload PDU from the IUT.

6. The Lower Tester keeps sending Nack to the IUT and the IUT resends the same payload PDU until transmit flush timeout. If encryption is enabled, continue on to step 7.

7. Repeat steps 1–6 except that only the second payload acknowledgement sends a CIS data PDU with empty payload.

**Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.10.2.11.1 LL/CIS/MAS/BV-13-C</strong></td>
<td>Disabled</td>
</tr>
<tr>
<td><strong>4.10.2.11.2 LL/CIS/MAS/BV-29-C</strong></td>
<td>Enabled</td>
</tr>
</tbody>
</table>

*Table 4.111: Acknowledgement Scheme Setup*

**Pass Verdict**

- In step 2, the IUT sends a payload PDU and in the PDU Header, both the SN and NESN fields match. The CIE and NPI fields have the value 0b0.

- In step 4, the IUT successfully uses the normal acknowledgement scheme repeating steps 2–3.

- In step 5, the IUT sends a new payload PDU, and in the Header, the SN field and the NESN field have the correct values per the Acknowledgement Scheme tables.

- If encryption is enabled, the previous pass criteria is met executing steps 1–6 as specified in step 7.

**4.10.2.12 LL/CIS/MAS/BV-17-C [Flushing of Packets in CIS, Master]**

**Test Purpose**

Test that the master IUT flushes data packets when the flush timeout has occurred.

**Reference**

[14] Section 4.5.13.5

**Initial Condition**

Connected in the master role as defined in the following initial state:

Note: "default" refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.112)
<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x09</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x05</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>

*Table 4.112: State Variable Values*
• Test Procedure

1. The Upper Tester submits payload data packets for the IUT to send to the Lower Tester.
2. The Lower Tester receives the first payload PDU from the IUT.
3. The Lower Tester does not send an Ack in response to any payload PDU received from the IUT.
4. The Lower Tester receives four of the same payload PDUs before subevent 5 elapses and the IUT payload is flushed. The Upper Tester observes that the FT endpoint is at the end of subevent 5 of the FT.
5. The Lower Tester receives a new payload PDU from the IUT at subevent 6.

• Expected Outcome

Pass Verdict
- In step 2, the IUT begins sending payload PDUs.
- In step 4, the IUT sends the first payload PDU four more times and flushes the PDU at the end of subevent 5 of the FT.
- In step 5, the IUT sends the second PDU at subevent 6 of the CIS Interval 1.

4.10.2.13  LL/CIS/MAS/BV-18-C [Bursting of Payloads in CIS, Master]

• Test Purpose
Test that the master IUT sends data payloads in a burst when Burst Number (BN) is greater than 1.

• Reference
[14] Section 4.5.13.3

• Initial Condition
Connected in the master role.
Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.113)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Table 4.113: State Variable Values
• Test Procedure

1. The Upper Tester submits two HCI ISO data packets for the IUT to send to the Lower Tester.
2. The Lower tester receives the first payload PDU and sends an Ack in response.
3. The Lower Tester receives the second payload on subevent 2 in CIS Interval 1 and does not send an Ack.
4. The Lower Tester receives the second payload on subevent 3 in CIS Interval 1 and does not send an Ack.
5. The Lower Tester receives the second payload on subevent 4 in CIS Interval 1 and sends an Ack.
6. The Upper Tester submits two new HCI ISO data packets for the IUT to send to the Lower Tester.
7. The Lower Tester receives payload three on subevent 1 in CIS Interval 2 and sends an Ack.

Figure 4.442: LL/CIS/MAS/BV-18-C [Bursting of Payloads in CIS, Master]
8. The Lower Tester receives payload four on subevent 2 in CIS Interval 2 and sends an Ack.
9. The Lower Tester does not receive any more payloads from the IUT.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends first payload on subevent 1 CIS Interval 1.
- In step 3, the IUT sends second payload on subevent 2 in CIS Interval 1.
- In step 4, the IUT sends the same payload in step 4 on subevent 3 in CIS Interval 1.
- In step 5, the IUT sends the same payload in step 4 on subevent 4 in CIS Interval 1.
- In step 7, the IUT sends third payload PDU on subevent 1 in CIS Interval 2.
- In step 8, the IUT sends fourth payload PDU on subevent 2 in CIS Interval 2.
- In step 9, the IUT does not send any more payloads to the Lower Tester.

4.10.2.14 LL/CIS/MAS/BV-19-C [Deterministic Packet Transmission in CIS, Master]

• Test Purpose
Test that the master IUT meets FLOOR (NSE/BN) number of transmissions for each data packet in the last channel interval before they are flushed.

• Reference
[14] Section 4.5.13.5

• Initial Condition
Connected in the master role as defined in the following initial state:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The latter values are specified in Section 4.10.1.2 Timing Requirements.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.114)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
</tbody>
</table>
## Table 4.114: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mx_sdu_m2s[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>
• Test Procedure

1. The Upper Tester submits the five HCI ISO data packets for the IUT to send to the Lower Tester.
2. The Lower Tester receives payload one on subevent 1 in CIS Interval 1 and sends an Ack in response.
3. The Lower Tester receives payload two on subevent 2 in CIS Interval 1, but doesn’t send an Ack.
4. The Lower Tester receives the same payload two on subevent 3 in CIS Interval 1, but doesn’t send an Ack.
5. The Lower Tester receives the same payload two on subevent 4 in CIS Interval 1, but doesn’t send an Ack.
6. The Lower Tester receives payload three on subevent 1 in CIS Interval 2, and sends a Nack.
7. The Lower Tester receives the same payload three on subevent 2 in CIS Interval 2, and sends a Nack.
8. The Lower Tester receives payload four on subevent 3 in CIS Interval 2, and sends a Nack.
9. The Lower Tester receives the same payload four on subevent 4 in CIS Interval 2, and sends a Nack.
10. The Lower Tester receives payload five on subevent 1 in CIS Interval 3 and sends an Ack.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends payload one on subevent 1 in CIS Interval 1.
- In step 3, the IUT sends payload two on subevent 2 in CIS Interval 1.
- In step 4, the IUT sends the same payload two on subevent 3 in CIS Interval 1.
- In step 5, the IUT sends the same payload two on subevent 4 in CIS Interval 1.
- In step 6, the IUT sends payload three on subevent 1 in CIS Interval 2.
- In step 7, the IUT sends the same payload three on subevent 2 in CIS Interval 2.
- In step 8, the IUT sends a new payload four on subevent 3 in CIS Interval 2.
- In step 9, the IUT sends the same payload four on subevent 4 in CIS Interval 2.
- In step 10, the IUT sends a new payload five on subevent 1 in CIS Interval 3.

4.10.3 SLA
Tests that the IUT behaves according to the CIS setup and CIS procedures as a slave.

4.10.3.1 [CIS Setup Response Procedure, Slave]

• Test Purpose
Test that a slave IUT sets up a CIS when requested by the master and maintains the connection in the slave role acknowledging data packets in the CIS.

• Reference
[14] Section 5.1.15

• Initial Condition
The Isochronous Channels (Host Support) FeatureSet bit is set.

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle on the PHY specified in Table 4.115: Slave Test Case Direction Specific Configurations.

The Lower Tester sets the parameters specified in Table 4.115: Slave Test Case Direction Specific Configurations and Table 4.116: Slave Test Case Additional Configurations. Any otherwise unspecified values are specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands, except for default values for cis_offset_mn and cis_offset_mx. The CIS offset values are specified in Section 4.10.1.2 Timing Requirements.
The Lower Tester is configured as the Master.

- **Test Procedure**

1. The Upper Tester sends an HCI_LE_Set_Event_Mask command with all events enabled, including the HCI_LE_CIS_Request event. The IUT sends a successful HCI_Command_Complete in response.
2. The Lower Tester sends an LL_CIS_REQ to the IUT with the contents specified in Table 4.115.
3. The Upper Tester receives an HCI_LE_CIS_Request event from the IUT and the parameters include CIS_Connection_Handle assigned by the IUT.
4. The Upper Tester sends an HCI_LE_Accept_CIS_Request command to the IUT, with the Connection_Handle field set to the value of the CIS_Connection_Handle received in step 3.
5. The Upper Tester expects the IUT to send a successful Command Status.
6. The Lower Tester receives an LL_CIS_RSP PDU from the IUT. In the message, the CIS_Offset_Min field and the CIS_Offset_Max field are equal to or a subset of the values received in the LL_CIS_REQ sent in step 2.
7. The Lower Tester sends an LL_CIS_IND where the CIS_Offset is the time (ms) from the start of the ACL connection event in connEvent Count to the first CIS anchor point, the CIS_Sync_Delay is CIG_Sync_Delay minus the offset from the CIG reference point to the CIS anchor point in μs, and the connEventCount is the CIS_Offset reference point.
8. The Upper Tester receives a successful HCI_LE_CIS_Established event from the IUT, after the first CIS packet sent by the LT. The Connection_Handle parameter is the CIS_Connection_Handle value provided in the HCI_LE_CIS_Request event.
9. The Lower Tester sends data packets to the IUT.
10. The Lower Tester receives an Ack.
11. Repeat steps 9 and 10 a total of 50 times.

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>M_To_S</th>
<th>Payload (PDU)</th>
<th>S_To_M</th>
<th>Payload (PDU)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PHY</strong></td>
<td><strong>BN</strong></td>
<td><strong>FT</strong></td>
<td><strong>PHY</strong></td>
<td><strong>BN</strong></td>
</tr>
<tr>
<td>LE 1M PHY</td>
<td>1</td>
<td>1</td>
<td>LE 1M PHY</td>
<td>1</td>
</tr>
<tr>
<td>LE 2M PHY</td>
<td>2</td>
<td>2</td>
<td>LE 2M PHY</td>
<td>2</td>
</tr>
<tr>
<td>LE 2M PHY</td>
<td>3</td>
<td>3</td>
<td>LE 2M PHY</td>
<td>3</td>
</tr>
<tr>
<td>LE 2M PHY</td>
<td>2</td>
<td>2</td>
<td>LE 2M PHY</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.115: Slave Test Case Direction Specific Configurations

<table>
<thead>
<tr>
<th>Test Case</th>
<th>NSE</th>
<th>SDU_Interval_M_to_S</th>
<th>SDU_Interval_S_To_M</th>
<th>ISO_Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CIS/SLA/BV-01-C</td>
<td>2</td>
<td>7.5 ms (0x1D4C)</td>
<td>7.5 ms (0x1D4C)</td>
<td>7.5 ms (0x06)</td>
</tr>
<tr>
<td>LL/CIS/SLA/BV-19-C</td>
<td>4</td>
<td>11.25 ms (0x2BF2)</td>
<td>11.25 ms (0x2BF2)</td>
<td>11.25 ms (0x09)</td>
</tr>
<tr>
<td>LL/CIS/SLA/BV-20-C</td>
<td>2</td>
<td>31.25 ms (0x7A12)</td>
<td>31.25 ms (0x7A12)</td>
<td>31.25 ms (0x10)</td>
</tr>
<tr>
<td>LL/CIS/SLA/BV-23-C</td>
<td>4</td>
<td>11.25 ms (0x2BF2)</td>
<td>11.25 ms (0x2BF2)</td>
<td>11.25 ms (0x09)</td>
</tr>
<tr>
<td>LL/CIS/SLA/BV-24-C</td>
<td>2</td>
<td>31.25 ms (0x7A12)</td>
<td>31.25 ms (0x7A12)</td>
<td>31.25 ms (0x10)</td>
</tr>
</tbody>
</table>

Table 4.116: Slave Test Case Additional Configurations

**Pass Verdict**
- In step 1, the IUT sends a successful HCI_Command_Complete.
- In step 3, the IUT sends an HCI_LE_CIS_Request event and the parameters include CIS_Connection_Handle assigned by the IUT.
- In step 5, the IUT sends a successful Command Status.
- In step 6, the IUT sends an LL_CIS_RSP PDU and in the message, the CIS_Offset_Min field and the CIS_Offset_Max field are equal to or a subset of the values received in the LL_CIS_REQ sent in step 1.
- In step 8, the IUT sends a successful HCI_LE_CIS_Established event. The Connection_Handle parameter is the CIS_Connection_Handle value provided in the HCI_LE_CIS_Request event.
- In step 11, the IUT acknowledges at least 48 of the data packets sent by the Lower Tester in the CIS.

4.10.3.2 LL/CIS/SLA/BV-02-C [CIS Setup Response Procedure, Slave, Reject Response]

- **Test Purpose**
  Test that a slave IUT rejects the request to set up a CIS.

- **Reference**
  [14] Section 5.1.15

- **Initial Condition**
  The Isochronous Channels (Host Support) FeatureSet bit is set.

  An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

  The Lower Tester acts in the master role.

- **Test Procedure**

  ![Diagram of the test procedure]

  **Figure 4.445: LL/CIS/SLA/BV-02-C [CIS Setup Procedure, Slave Responding, Reject Response]**

  1. The Upper Tester sends an HCI_LE_Set_Event_Mask command with all events enabled, including the HCI_LE_CIS_Request event. The IUT sends a successful HCI_Command_Complete in response.
  2. The Lower Tester sends an LL_CIS_REQ PDU with valid data as specified for the Set CIG Parameters command in Section 4.10.1.3 Default Values for Set CIG Parameters Commands to the IUT.
  3. The Upper Tester receives an HCI_LE_CIS_Request event from the IUT.
  4. The Upper Tester sends an HCI_LE_Reject_CIS_Request command to the IUT with a valid reason code and receives a successful return status.
5. The Upper Tester receives an HCI_Command_Complete event from the IUT.
6. The Lower Tester receives an LL_REJECT_EXT_IND from the IUT with a valid reason code.

- **Expected Outcome**
  - **Pass Verdict**
  - In step 1, the IUT sends a successful HCI_Command_Complete.
  - In step 3, the IUT sends an HCI_LE_CIS_Request event.
  - In step 4, the IUT sends a successful return status.
  - In step 5, the IUT sends an HCI_Command_Complete event.
  - In step 6, the IUT sends an LL_REJECT_EXT_IND with a valid reason code.

4.10.3.3 LL/CIS/SLA/BV-03-C [CIS Map Update]

- **Test Purpose**
  Tests that the IUT updates the channel map for a CIS when the channel map for the connection is updated.

- **Reference**
  [16] Section 6.3
  [14] Sections 5.1.2, 2.4.2.2

- **Initial Condition**
  A CIG has been established.
  The initial channel map is 0x1FFFFFFFFF (i.e., all channels enabled).
  The Lower Tester acts in the master role.

- **Test Procedure**

  ![Diagram of test procedure]

  **Figure 4.446: LL/CIS/SLA/BV-03-C [CIS Map Update]**

  1. The Lower Tester sends an LL_CHANNEL_MAP_IND command to the IUT with the ChM field in CtrData field value 0x1249249249 and a valid instant field value.
  2. After the instant has passed, the Upper Tester sends an HCI_LE_Read_Channel_Map command to IUT.
3. The Lower Tester expects that the return parameter Channel_Map received has value 0x1249249249.
4. The Lower Tester sends data packets to the IUT.
5. The IUT sends an ACK to the Lower Tester.
6. Repeat steps 4 and 5 for 50 total events.

- Expected Outcome
  
  Pass Verdict
  
  - In step 3, the IUT sends return parameter with value 0x1249249249.
  - In step 5, the IUT sends acks for at least 48 of the data packets received from the Lower Tester.

4.10.3.4 LL/CIS/SLA/BV-22-C [CIS Request Event Not Set]

- Test Purpose
  Confirm the IUT does not generate an HCI_LE_CIS_Request event when the event is masked.

- Reference
  [15] Section 7.7.65.26

- Initial Condition
  The Isochronous Channels (Host Support) FeatureSet bit is clear.
  An ACL connection has been established between the IUT and Lower Tester with the IUT acting as the Slave.
• Test Procedure

1. The Lower Tester sends an LL_CIS_REQ to the IUT with the contents specified per Section 4.10.1.3 Default Values for Set CIG Parameters Commands.
2. The IUT responds to the Lower Tester with an LL_REJECT_EXT_IND with error code Unsupported Remote Feature / Unsupported LMP Feature (0x1A).
3. The IUT disconnects the ACL connection from the Lower Tester.
4. The Upper Tester sends an HCI_LE_Set_Host_Feature command to the IUT with the Bit_Number set to 32 (Isochronous Channels) and the Bit_Value set to 0b1. The Upper Tester receives an HCI_Command_Complete event from the IUT.
5. The IUT establishes an ACL connection with the Lower Tester as Slave.
6. The Upper Tester sends an HCI_LE_Set_Event_Mask command with all events enabled except the HCI_LE_CIS_Request event. The IUT sends a successful HCI_Command_Complete in response.
7. Repeat step 1.
8. The IUT responds to the Lower Tester with an LL_REJECT_EXT_IND with a valid reason code.

Figure 4.447: LL/CIS/SLA/BV-22-C [CIS Request Event Not Set]
9. The Upper Tester does not receive an HCI_LE_CIS_Request event. Confirm for at least 5 seconds.
10. The Upper Tester sends an HCI_LE_Set_Event_Mask command with all events enabled including the HCI_LE_CIS_Request event. The IUT sends a successful HCI_Command_Complete in response.
11. Repeat step 1.
12. The Upper Tester receives an HCI_LE_CIS_Request event.

- Expected Outcome

Pass Verdict
- In step 2, the IUT responds with LL_REJECT_EXT_IND and error code Unsupported Remote Feature / Unsupported LMP Feature (0x1A).
- In steps 4, 6, and 10, the IUT sends a successful HCI_Command_Complete.
- In step 9, the Upper Tester does not receive an HCI_LE_CIS_Request event.
- In step 12, the Upper Tester receives an HCI_LE_CIS_Request event.

4.10.3.5 LL/CIS/SLA/BV-07-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG, Slave]

- Test Purpose
Test that the slave IUT can send and receive data on multiple CISes in an interleaved arrangement on a single CIG on a single connection.

- Reference
[14] Section 4.5.14.2

- Initial Condition
Connected in the slave role as defined in the following initial state:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Slave (values as specified in Table 4.117)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x01 (Interleaved)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>2</td>
</tr>
</tbody>
</table>
### State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.117: State Variable Values*
1. The Upper Tester Sends four HCI ISO Data Packets to the IUT, two for each CIS.
2. The Lower Tester sends a Null PDU to the IUT on CIS(1).
3. The IUT sends the first CIS(1) payload to the Lower Tester on CIS(1). Note the CIS_ID of CIS(1).
4. The Lower Tester sends a Null PDU to the IUT on CIS(2).
5. The IUT sends the first CIS(2) payload to the Lower Tester on CIS(2). Note the CIS_ID of CIS(2).
6. The Lower Tester sends its first CIS(1) payload to the IUT on CIS(1).
7. The IUT receives the PDU 1 from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
8. The IUT sends the second CIS(1) payload to the Lower Tester on CIS(1).
9. The Lower Tester sends its first CIS(2) payload to the IUT on CIS(2).
10. The IUT receives the PDU 1 from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
11. The IUT sends the second CIS(2) payload to the Lower Tester on CIS(2).
12. The Lower Tester sends its second CIS(1) payload on CIS(1).
13. The IUT receives the PDU 2 from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
14. The IUT sends an Ack on CIS(1).
15. The Lower Tester sends its second CIS(2) payload to the Lower Tester on CIS(2).
16. The IUT receives the PDU 2 from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
17. The IUT sends an Ack to the Lower Tester on CIS(2).
18. Repeat all prior steps 3 times.

- Expected Outcome

**Pass Verdict**

- In step 3, the IUT sends a CIS payload to the Lower Tester on CIS(1). The payload is one of the first payloads provided by the Upper Tester for a CIS to the IUT.
- In step 5, the IUT sends a CIS payload to the Lower Tester on CIS(2). The payload is the other first payload provided by the Upper Tester for a CIS to the IUT.
- In step 7, the IUT receives the PDU from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
- In step 8, the IUT sends a second CIS payload to the Lower Tester on CIS(1). The payload is the second payload provided by the Upper Tester for the CIS with the CIS_ID in step 3.
- In step 10, the IUT receives the PDU from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
- In step 11, the IUT sends a second CIS payload to the Lower Tester on CIS(2). The payload is the second payload provided by the Upper Tester for the CIS with the CIS_ID in step 5.
- In step 13, the IUT receives the PDU from the Lower Tester on CIS(1). The IUT sends the payload to the Upper Tester.
- In step 14, the IUT sends an Ack on CIS(1).
- In step 16, the IUT receives the PDU from the Lower Tester on CIS(2). The IUT sends the payload to the Upper Tester.
- In step 17, the IUT sends an Ack to the Lower Tester on CIS(2).
- The prior results in steps 1–17 repeat 3 times.

**4.10.3.6 LL/CIS/SLA/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential, Slave]**

- **Test Purpose**
  Test that the slave IUT can send and receive data on multiple CISes in a sequential arrangement in a single Connected Isochronous Group on a single connection.
• Reference
  
  [14] Section 4.5.14.2

• Initial Condition
  
  Connected in the slave role as defined in the following initial state:

  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

  State: Connected Isochronous Stream, Slave (values as specified in Table 4.118)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>2</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.118: State Variable Values*
• Test Procedure

1. The Upper Tester orders the IUT to send two payload PDUs to the Lower Tester to be sent on CIS(1).
2. The Lower Tester sends a payload PDU to the IUT on CIS(1).
3. The IUT sends a payload PDU and an Ack on CIS(1) to the Lower Tester T_IFS after receiving the Lower Tester’s payload PDU.
4. The Lower Tester sends an Ack on CIS(1).
5. The IUT sends a payload PDU and an Ack on CIS(1) to the Lower Tester T_IFS after receiving the Lower Tester’s Ack.
6. When CIS(1) interval ends, repeat steps 1–5 on CIS(2).
7. Repeat steps 1–6.

Figure 4.449: LL/CIS/SLA/BV-08-C [Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential, Slave]
• Expected Outcome

Pass Verdict
- In step 3, the IUT sends a payload PDU at CIS(1) and Ack the Lower Tester's payload.
- In step 5, the IUT sends a payload PDU and an Ack at CIS(1)'s subevent 2.
- In step 6, the IUT sends the payload PDU on CIS(2) and sends the second payload PDU at CIS(2)'s subevent 2.
- In steps 1–6, the results repeat.

4.10.3.7 [Acknowledgement Scheme, Slave]

• Test Purpose
Test that the slave IUT can use the acknowledgement scheme.

• Reference
[14] Section 4.5.9

• Initial Condition
Connected in the slave role.

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Slave (values as specified in Table 4.119)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdv_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdv_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>0x02</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>0x02</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
</tbody>
</table>
### State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

Table 4.119: State Variable Values

- **Test Procedure**

1. The Upper Tester orders the IUT to send four separate payload PDUs to the Lower Tester.
2. The Lower Tester sends a PDU to the IUT as follows:
   a. On the first subevent, it sends a CIS data PDU with empty payload to the IUT.
   b. After the first subevent, it sends an Ack to the IUT.
3. The Lower Tester receives a payload PDU from the IUT. The IUT observes the acknowledgement scheme by using the next SN for each payload PDU sent where the NESN in the packet received is the next SN and by using the next NESN where the current NESN matches the SN in the PDU correctly received.

![Diagram of test procedure](image-url)
4. Repeat steps 2–3 for the second and third payload PDUs, except the Lower Tester acknowledges the payload but does not send a CIS data PDU with empty payload.

5. The Lower Tester receives the fourth payload PDU from the IUT. The Lower Tester sends an Ack before receiving the fourth payload.

6. The Lower Tester keeps sending Nack to the IUT and the IUT resends the same payload PDU until transmit flush timeout.

If encryption is enabled, continue on to step 7.

7. Repeat steps 1–6 except that only the second payload acknowledgement sends a CIS data PDU with empty payload.

- Expected Outcome

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.10.3.7.1 LL/CIS/SLA/BV-10-C</td>
<td>Disabled</td>
</tr>
<tr>
<td>4.10.3.7.2 LL/CIS/SLA/BV-21-C</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Table 4.120: Acknowledgement Scheme Setup

Pass Verdict
- In step 3, the IUT sends a payload PDU and in the PDU Header, both the SN and NESN fields match. The CIE and NPI fields have the value 0b0.
- In step 4, the IUT successfully uses the normal acknowledgement scheme repeating steps 2–3.
- In step 5, the IUT sends a new payload PDU, and in the Header, the SN field and the NESN field have the correct values per the Acknowledgement Scheme tables.
- If encryption is enabled, the previous pass criteria is met executing steps 1–6 as specified in step 7.

4.10.3.8 LL/CIS/SLA/BV-14-C [Flushing of Packets in CIS, Slave]

- Test Purpose
  Test that the IUT flushes data packets when the flush timeout has occurred, and the packets are not acknowledged by the Lower Tester.

- Reference
  [14] Section 4.5.13.5

- Initial Condition
  Connected in the slave role as defined in the following initial state:
  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Slave (values as specified in Table 4.121)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x09</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>default</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x05</td>
</tr>
</tbody>
</table>

*Table 4.121: State Variable Values*
• Test Procedure

1. The Upper Tester submits payload data packets for the IUT to send to the Lower Tester.
2. Lower Tester sends a Null PDU to the IUT. The Lower Tester then receives the first payload PDU from the IUT.
3. The Lower Tester sends a NACK in response to any payload PDU received from the IUT.
4. The Lower Tester receives four of the same payload PDUs before subevent 5 elapses and the IUT payload is flushed. The Upper Tester observes that the FT endpoint is at the end of subevent 5 of the FT.
5. The Lower Tester receives a new payload PDU from the IUT at subevent 6.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends a payload PDU.
- In step 4, the IUT sends the first payload PDU four more times and flushes the PDU at the end of subevent 5 of the FT.
- In step 5, the IUT sends the second PDU at subevent 6 of the CIS Interval 1.

4.10.3.9 LL/CIS/SLA/BV-15-C [Bursting of Payloads in CIS, Slave]

• Test Purpose
Test that the slave IUT sends data payloads in a burst when Burst Number (BN) is greater than 1.

• Reference
[14] Section 4.5.13.3

• Initial Condition
Connected in the slave role.
Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Slave (values as specified in Table 4.122)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.122: State Variable Values*
• Test Procedure

1. The Upper Tester submits two HCI ISO data packets for the IUT to send to the Lower Tester.
2. The Lower sends a Null PDU on subevent 1 in CIS Interval 1 and in response the IUT sends Payload 1.
3. The Lower Tester sends an Ack in subevent 2 in CIS Interval 1 and in response the IUT sends the second payload.
4. The Lower Tester sends a Nack in subevent 3 in CIS Interval 1 and in response the IUT resends the second payload.

Figure 4.452: LL/CIS/SLA/BV-15-C [Bursting of Payloads in CIS, Slave]
5. The Lower Tester sends a Nack in subevent 4 in CIS Interval 1 and in response the IUT resends the second payload.
6. The Upper Tester submits two new HCI ISO data packets for the IUT to send to the Lower Tester.
7. The Lower Tester sends a Null PDU on subevent 1 in CIS Interval 2 and in response the IUT sends payload three.
8. The Lower Tester sends an Ack on subevent 2 in CIS Interval 2 and in response the IUT sends payload four.
9. The Lower Tester sends an Ack on subevent 3 in CIS Interval 2.
10. The Lower Tester does not receive any more payloads from the IUT.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends first payload on subevent 1 CIS Interval 1.
- In step 3, the IUT sends second payload on subevent 2 in CIS Interval 1.
- In step 4, the IUT sends the same payload in step 4 on subevent 3 in CIS Interval 1.
- In step 5, the IUT sends the same payload in step 4 on subevent 4 in CIS Interval 1.
- In step 7, the IUT sends third payload PDU on subevent 1 in CIS Interval 2.
- In step 8, the IUT sends fourth payload PDU on subevent 2 in CIS Interval 2.
- In step 10, the IUT does not send any more payloads to the Lower Tester.

4.10.3.10 LL/CIS/SLA/BV-16-C [Deterministic Packet Transmission in CIS, Slave]

• Test Purpose
Test that the slave IUT meets FLOOR (NSE/BN) number of transmissions for each data packet in the last channel interval before they are flushed.

• Reference
[14] Section 4.5.13.3

• Initial Condition
Connected in the slave role as defined in the following initial state:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Slave (values as specified in Table 4.123)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>default</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.123: State Variable Values*
• Test Procedure

1. The Upper Tester submits the five HCI ISO data packets for the IUT to send to the Lower Tester.
2. The Lower Tester sends a Null PDU to the IUT. The Lower Tester receives payload one on subevent 1 in CIS Interval 1 and sends an Ack in response.
3. The Lower Tester receives payload two on subevent 2 in CIS Interval 1, but it sends a Nack to the IUT.
4. The Lower Tester receives the same payload two on subevent 3 in CIS Interval 1, but it sends a Nack to the IUT.
5. The Lower Tester receives the same payload two on subevent 4 in CIS Interval 1.
6. The Lower Tester sends a Null PDU. The Lower Tester receives payload three on subevent 1 in CIS Interval 2, and sends a Nack.
7. The Lower Tester receives the same payload three on subevent 2 in CIS Interval 2, and sends a Nack.
8. The Lower Tester receives payload four on subevent 3 in CIS Interval 2, and sends a Nack.
9. The Lower Tester receives the same payload four on subevent 4 in CIS Interval 2.
10. The Lower Tester sends a Null PDU. The Lower Tester receives a payload five on subevent 1 in CIS Interval 3 and sends an Ack.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends payload one on subevent 1 in CIS Interval 1.
- In step 3, the IUT sends payload two on subevent 2 in CIS Interval 1.
- In step 4, the IUT sends the same payload two on subevent 3 in CIS Interval 1.
- In step 5, the IUT sends the same payload two on subevent 4 in CIS Interval 1.
- In step 6, the IUT sends payload three on subevent 1 in CIS Interval 2.
- In step 7, the IUT sends the same payload three on subevent 2 in CIS Interval 2.
- In step 8, the IUT sends a new payload four on subevent 3 in CIS Interval 2.
- In step 10, the IUT sends the same payload four on subevent 4 in CIS Interval 2.
- In step 11, the IUT sends a new payload five on subevent 1 in CIS Interval 3.

4.10.4 Both Connected Roles

4.10.4.1 [Sending data in Unidirectional CIS]

• Test Purpose
Test that the IUT can send data to a device in the Connected Isochronous Group.

• Test Case IDs

LL/CIS/MAS/BV-05-C
LL/CIS/SLA/BV-04-C

• Reference

[14] Section 4.5.13, 4.5.13.3

• Initial Condition

Connected in the relevant role as defined in the following initial states:

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.124)
### Table 4.124: State Variable Values

State: Connected Isochronous Stream, Slave (values as specified in Table 4.125)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x88B8 (35 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x88B8 (35 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x54 (105 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x06</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x03</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x06</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x03</td>
</tr>
</tbody>
</table>

*Table 4.125: State Variable Values*
• Test Procedure

1. The Upper Tester orders the IUT to send three separate payload PDUs to the Lower Tester.
2. The Lower Tester receives the first payload (PDU 0) on subevent n. The size in octets of the PDU received doesn’t exceed the Payload_M_To_S in LL_CIS_REQ when the CIS is created. In the PDU header, the LLID field contains the Logical Link Identifier and has value 0b00, and the CIE field contains the Close Isochronous event bit and has value 0b0, the NPI field has value 0b0, and the Length field contains the size of the payload in octets in range of 0 to 255.
   a. If the IUT is in the slave role, the Lower Tester sends a NULL PDU before it receives the first payload.
   b. If the IUT is in the master role, the Lower Tester sends a NULL PDU after it receives the first payload.
3. The IUT sends the second payload (PDU 1) to the Lower Tester.
   a. If the IUT is in the slave role, the Lower Tester sends a NULL PDU before it receives the second payload.
   b. If the IUT is in the master role, the Lower Tester sends a NULL PDU after it receives the first payload.
4. The Lower Tester receives the second payload (PDU 1) from the IUT on the subevent n+1. In the PDU header, the LLID field is 0b00, and the CIE field is 0b0.
5. The IUT sends the third payload PDU to the Lower Tester. The Lower Tester receives the new payload (PDU 2) from the IUT on the subevent n+2. In the PDU header, the LLID field is 0b00, the NESN and SN fields are both 0b0, and the CIE field is 0b0.
   a. If the IUT is in the slave role, the Lower Tester sends a NULL PDU before the IUT sends the new payload (PDU 2).
6. The Lower Tester does not Ack the payload (PDU 2) transmission if the IUT is Master, and Nacks the payload if the IUT is Slave. The Lower Tester receives the same payload (PDU 2) from the IUT on the next subevent.

- **Expected Outcome**
  - **Pass Verdict**
    - In step 2, the IUT sends the first payload PDU. The size in octets of the payload in the payload PDU received doesn't exceed the Frame_Size_M_To_S in LL_CIS_REQ when the CIS is created.
    - In step 2, the LT validates the LLID field, the NESN field, the SN field, and the CIE field in the PDU header.
    - In step 3, the IUT sends the second payload PDU 1 on the next subevent and in the PDU header and the LLID field, the NESN field, the SN field, and the CIE field have the values as specified in step 4.
    - In step 4, the LT validates the LLID field, the NESN field, the SN field, and the CIE field in the PDU header.
    - In step 5, the IUT sends the third payload PDU 2 on the n+2 subevent. In the PDU header, the LLID field is 0b00, the NESN and SN fields are both 0b0, and the CIE field is 0b0.
    - In step 6, the IUT sends the same payload PDU 2 on the next subevent.

4.10.4.2 [Receiving data in Unidirectional CIS]

- **Test Purpose**
  Test that the IUT can receive data sent by a device in the CIS.

- **Test Case IDs**
  LL/CIS/MAS/BV-06-C
  LL/CIS/SLA/BV-05-C

- **Reference**
  [14] Section 4.5.13, 4.5.13.3

- **Initial Condition**
  Connected in the relevant role as defined in the following initial states:
  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.
State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.126)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Table 4.126: State Variable Values

State: Connected Isochronous Stream, Slave (values as specified in Table 4.127)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.127: State Variable Values*
• **Test Procedure**

![Diagram](image)

**Figure 4.455: Receiving data in Unidirectional CIS**

1. The Lower Tester is set to send three separate payload PDUs to the IUT.
2. The Lower Tester sends the first payload (PDU 0) on subevent n. The size in octets of the PDU received doesn't exceed the max payload size sent in LL_CIS_REQ when the CIS is created. In the PDU header, the LLID field contains the Logical Link Identifier and has value 0b00, and the CIE field contains the Close Isochronous Event bit and has value 0b0, the NPI field has value 0b0, and the Length field contains the size of the payload in octets in range of 0 to 255.
   a. If the IUT is in the master role, the LT sends a NULL PDU after the LT receives the payload.
   b. If the IUT is in the slave role, the LT sends a NULL PDU before the IUT sends the payload.
3. The IUT sends the received ISO Data to the Upper Tester.
4. The Lower Tester sends the second payload PDU to the IUT.
5. The IUT receives the second payload (PDU 1) from the Lower Tester on the subevent n+1. In the PDU header, the LLID field is 0b00, and the CIE field is 0b0.
   a. If the IUT is in the master role, the LT sends a NULL PDU after the LT receives the payload.
b. If the IUT is in the slave role, the LT sends a NULL PDU before the IUT sends the payload.

6. The Lower Tester sends the third PDU (PDU 2) to the IUT. The IUT receives the third payload (PDU 2) from the Lower Tester on the subevent 1 in CIS Interval 2. In the PDU header, the LLID field is 0b00, the NESN and SN fields are both 0b0, and the CIE field is 0b0.

• Expected Outcome

**Pass Verdict**

- In step 2, the Lower Tester sends the first payload PDU. The size in octets of the payload in the payload PDU received doesn't exceed the maximum payload sent in LL_CIS_REQ when the CIS is created.

- In step 2, the IUT validates the LLID field, the NESN field, the SN field, and the CIE field in the PDU header.

- In step 5, the IUT receives the second payload PDU on the second subevent and in the PDU header and the LLID field, the NESN field, the SN field, and the CIE field have the values as specified in step 6.

- In step 6, the IUT receives a new payload PDU 2 on the first subevent in Interval 2. In the PDU header, the LLID field is 0b00, the NESN and SN fields are both 0b0, and the CIE field is 0b0.

**4.10.4.3 [Sending and Receiving Data in Bidirectional CIS]**

• Test Purpose

Test that the IUT can send and receive data on a CIS.

• Test Case IDs

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL/CIS/MAS/BV-07-C</td>
<td>Test send and receive data in bidirectional CIS.</td>
</tr>
<tr>
<td>LL/CIS/SLA/BV-06-C</td>
<td>Test send and receive data in bidirectional CIS.</td>
</tr>
</tbody>
</table>

• Reference

[14] Section 4.5.13, 4.5.13.3

• Initial Condition

Connected in the relevant role as defined in the following initial states:

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.128)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0  (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0  (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table 4.128: State Variable Values

State: Connected Isochronous Stream, Slave (values as specified in Table 4.129)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x06</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>251</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x03</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x186A0 (100 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x50 (100 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x06</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>251</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>251</td>
</tr>
</tbody>
</table>
### Test Procedure

1. The Upper Tester submits three separate payload data packets for the IUT to send to the Lower Tester. Note: If the role of the IUT is master, proceed to step 2 then step 3. If the role of the IUT is slave, proceed with step 3 then step 2.

2. The Lower Tester receives the first payload PDU.
   a. When IUT is master, the size in octets of the PDU received doesn’t exceed the Payload_M_To_S in LL_CIS_REQ when the CIS is created. In the PDU header, the LLID field has the value 0b00, the NESN and SN fields both have the value 0b0, and the CIE field has the value 0b0. If the CIE field has the value 0b1, then the Lower Tester listens to the IUT in the following subevent to make sure that the IUT does not transmit any further packets in that event.
   b. When IUT is slave, the size in octets of the PDU received doesn’t exceed the Payload_S_To_M in LL_CIS_REQ when the CIS is created. In the PDU header, the LLID field has the value 0b00, the NESN and SN fields both have the value 0b0, and the CIE field has the value 0b0. If the CIE field has the value 0b1, then the Lower Tester transmits a packet to the IUT in the following subevent to make sure that the IUT does not transmit any further packets in that event.
3. The Lower Tester sends a payload PDU.
   a. To IUT T_IFS after receiving the PDU. In the PDU header, the LLID field is set to 0b00, the CIE field is set to 0b0, and the NESN and SN fields set to the values according to the Acknowledgement Scheme specified in Common Variables when IUT is master.
   b. To IUT after receiving the PDU at the CIS anchor point. In the PDU header, the LLID field is set to 0b00, the CIE field is set to 0b0, and the NESN and SN fields set to the values according to the Acknowledgement Scheme specified in Common Variables when IUT is slave.
4. The Upper Tester observes that the IUT receives the PDU and reports the received data to the host.
5. Steps 2–4 are repeated 2 more times for the second and third payload PDUs. The formula for the Sub_Interval can vary.
   a. For the second payload PDU, the subevent 1 interval between the start of subevent 1 and the start of subevent 2 is at Sub_interval + drift (see [14] Section 4.5.13.1 for calculation).
   b. For the third payload PDU, subevent 2 interval between the start of subevent 2 and the start of subevent 3 is at Sub_interval + drift. The CIE field has the value 0b0.

- **Expected Outcome**

  **Pass Verdict**

  - In step 2, the size in octets of the payload in the payload PDU received doesn't exceed the Payload_Size_M_To_S in LL_CIS_REQ when the CIS is created when the IUT is master. The size in octets of the payload in the payload PDU received doesn't exceed the Payload_Size_M_To_S in LL_CIS_REQ when the CIS is created when the IUT is slave. If the CIE field has the value 0b1, then the IUT does not transmit any further packets in that event.
  - In step 2, the Lower Tester validates that in the PDU header, the LLID field has the value 0b00, the NESN and SN fields both have the value 0b0, and the CIE field has the value 0b0. If the CIE is set to the value 0b1, the IUT does not transmit any further packets in that event.
  - In step 4, the IUT reports the received data to the host, and in the HCI Isochronous Data packet, the Burst_Flag value is 0b0.
  - In step 5a, the IUT sends the second payload PDU on the second subevent. In the PDU header, the LLID field has the value 0b00, and the NESN and SN fields have the values according to the Acknowledgement Scheme. The subevent 1 interval between the start of subevent 1 and the start of subevent 2 is the calculated value of Sub_Interval. If the CIE field has the value 0b1, then in step 10, the IUT won't send a PDU. The IUT sends the received data to the host and in the HCI Isochronous Data packet, the Burst_Flag value is 0b01.
  - In step 5b, the IUT sends the third payload PDU on the third subevent. In the PDU header, the LLID field has the value 0b00, and the NESN and SN fields set to the values according to the Acknowledgement Scheme. The subevent 2 interval between the start of subevent 2 and the start of subevent 3 is between Sub_Interval_Min and Sub_Interval, inclusive. The IUT sends the received data to the host and in the HCI Isochronous Data packet, the Burst_Flag value is 0b10. The CIE field has the value 0b0.

4.10.4.4 [Retransmissions]

- **Test Purpose**
  Test that the IUT correctly uses the acknowledgment scheme and retransmits a data packet on a negative acknowledgment.
• Test Case IDs
  LL/CIS/MAS/BV-14-C
  LL/CIS/SLA/BV-11-C

• Reference
  [14] Section 4.5.9

• Initial Condition
  Connected in the relevant role as defined in the following initial states:
  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.
  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.130)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>0x02</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>0x02</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x02</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>0</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Table 4.130: State Variable Values
State: Connected Isochronous Stream, Slave (values as specified in Table 4.131)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>0x02</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>0x02</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x02</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>0</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x00</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x01</td>
</tr>
</tbody>
</table>

*Table 4.131: State Variable Values*
• Test Procedure

1. The Upper Tester submits two data packets for the IUT to send to the Lower Tester.
2. The Lower Tester receives the first payload PDU from the IUT.
3. The Lower Tester does not send an Ack or sends a Nack in response to the payload PDU in step 2.
4. The Lower Tester receives the same payload PDU with the same SN and NESN fields in the subevent 2 of the CIS Interval 1.
5. The Lower Tester sends an Ack after receiving the payload PDU in step 4.
6. The Lower Tester receives the second payload PDU from the IUT at subevent 1 of the CIS Interval 2.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends the first payload PDU.
- In step 4, the IUT sends the first payload PDU on subevent 2 of CIS Interval 1.
- In step 5, the IUT sends the second payload PDU at the subevent 1 of CIS Interval 2.

4.10.4.5 [CIS Terminate Procedure, Initiated]

• Test Purpose

Test that the IUT can terminate a CIS by sending the termination packet. Also test the error case for the HCI_LE_Remove_CIG command.
• Test Case IDs
  LL/CIS/MAS/BV-15-C
  LL/CIS/SLA/BV-12-C

• Reference
  [14] Section 2.4.2.3, 5.1.16
  [15] Section 7.8.100

• Initial Condition
  Connected in the relevant role as defined in the following initial states:

  Note: "default" refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.132)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

Table 4.132: State Variable Values
State: Connected Isochronous Stream, Slave (values as specified in Table 4.133)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

*Table 4.133: State Variable Values*
• Test Procedure

1. A payload PDU and Ack is sent between the IUT and Lower Tester.
2. The Upper Tester sends an HCI_LE_Remove_CIG command with the CIG_ID value obtained when the CIG was established.
3. The Upper Tester receives an error code Command Disallowed (0x0C).
4. The Upper Tester sends an HCI_Disconnect command to the IUT with a valid Connection_Handle and Reason code and receives a successful HCI_Command_Status event from the IUT in response.
5. The Lower Tester receives an LL_CIS_TERMINATE_IND PDU from the IUT and the ErrorCode field in the CtrData field matches the Reason code value the Upper Tester sent in step 4.
6. The Lower Tester sends an Ack to the IUT.
7. The Upper Tester receives an HCI_Disconnection_Complete event from the IUT.

• Expected Outcome

Pass Verdict
- In step 3, the IUT sends error code Command Disallowed (0x0C) to the Upper Tester.
  - If the IUT is in slave role, the error code is sent.
  - If the IUT is in master role and the CIG has one or more CISes established.
- In step 4, the IUT sends the Upper Tester a successful HCI_Command_Status event.
- In step 5, the IUT sends an LL_CIS_TERMINATE_IND PDU and the ErrorCode field in the CtrData field matches the Reason code value in step 5.
- In step 6, the IUT receives an Ack from the Lower Tester.
- In step 7, the Upper Tester receives an HCI_Disconnection_Complete event from the IUT.
4.10.4.6 [CIS Terminate Procedure, Accepting]

- **Test Purpose**
  Test that the IUT accepts CIS termination PDU from the Lower Tester.

- **Test Case IDs**
  LL/CIS/MAS/BV-16-C
  LL/CIS/SLA/BV-13-C

- **Reference**
  [14] Section 2.4.2.32, 5.1.16
  [15] Section 7.7.5

- **Initial Condition**
  Connected in the relevant role as defined in the following initial states:

  Note: “default” refers to values specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands.

  State: Connected Isochronous Stream, Master, Test (values as specified in Table 4.134)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
</tbody>
</table>
### Table 4.134: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>

State: Connected Isochronous Stream, Slave (values as specified in Table 4.135)

### Table 4.135: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdu_int_m2s</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>sdu_int_s2m</td>
<td>0x4E20 (20 ms)</td>
</tr>
<tr>
<td>ft_m2s</td>
<td>1</td>
</tr>
<tr>
<td>ft_s2m</td>
<td>1</td>
</tr>
<tr>
<td>iso_int</td>
<td>0x10 (20 ms)</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>cis_cnt</td>
<td>1</td>
</tr>
<tr>
<td>nse[]</td>
<td>0x04</td>
</tr>
<tr>
<td>mx_sdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_sdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_m2s[]</td>
<td>130</td>
</tr>
<tr>
<td>mx_pdu_s2m[]</td>
<td>130</td>
</tr>
<tr>
<td>phy_m2s[]</td>
<td>0x01</td>
</tr>
<tr>
<td>phy_s2m[]</td>
<td>0x01</td>
</tr>
<tr>
<td>bn_m2s[]</td>
<td>0x02</td>
</tr>
<tr>
<td>bn_s2m[]</td>
<td>0x02</td>
</tr>
</tbody>
</table>
• Test Procedure

![Diagram of test procedure]

Figure 4.459: CIS Terminate Procedure, Accepting

1. A payload PDU and Ack is sent between the IUT and Lower Tester.
2. The Lower Tester sends an LL_CIS_TERMINATE_IND PDU to the IUT and receives an Ack from the IUT.
3. The Upper Tester receives an HCI_Disconnection_Complete event from the IUT.

• Expected Outcome

Pass Verdict
- In step 2, the IUT sends an Ack.
- In step 3, the IUT sends an HCI_Disconnection_Complete event to the Upper Tester.
- The Lower Tester does not receive any payload PDUs from the IUT after step 3.

4.10.4.7 [CIS Updating Peer Clock Accuracy]

• Test Purpose
Verify that an IUT can update the Peer Clock Accuracy.

• Test Case IDs
"LL/CIS/MAS/BV-24-C"
"LL/CIS/SLA/BV-18-C"

• Reference
[14] Section 5.1.14
[15] Section 7.8.108

• Initial Condition
Connected in the relevant role.
An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.
• **Test Procedure**

![Figure 4.460: CIS Updating SCA](image)

1. The Upper Tester sends an `HCI_LE_Request_Peer_SCA` command to the IUT with a valid `Connection_Handle`.
2. The Upper Tester receives the Command Status event.
3. The Lower Tester receives an `LL_CLOCK_ACCURACY_REQ` PDU from the IUT and sends back an `LL_CLOCK_ACCURACY_RSP` PDU with a valid `Peer_Clock_Accuracy` value.
4. The Upper Tester receives an `HCI_LE_Request_Peer_SCA_Complete` event with the `Connection_Handle` of the ACL and the `Peer_Clock_Accuracy` parameter when the `HCI_LE_Request_Peer_SCA` command completes.
5. The Upper Tester sends an `HCI_LE_Request_Peer_SCA` command to the IUT with a valid, but non-existent `Connection_Handle` and receives the error code Unknown Connection Identifier (0x02).

• **Expected Outcome**

**Pass Verdict**
- In step 2, the IUT sends a Command Status event.
- In step 3, the IUT sends an `LL_CLOCK_ACCURACY_REQ` command.
- In step 4, the IUT sends an `HCI_LE_Request_Peer_SCA_Complete` event with the `Connection_Handle` of the ACL and the `Peer_Clock_Accuracy` parameter.
- In step 5, the IUT sends the error code Unknown Connection Identifier (0x02).

4.10.4.8 **[Error Updating Peer Clock Accuracy]**

• **Test Purpose**

Verify that the IUT returns an error when the Upper Tester requests the `Peer_Clock_Accuracy` to an IUT that doesn’t support Peer SCA Reporting.
• Test Case IDs
  LL/CIS/MAS/BI-01-C
  LL/CIS/SLA/BI-01-C

• Reference
  [14] Section 5.1.14
  [15] Section 7.8.108

• Initial Condition
  Connected in the relevant role.
  An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

• Test Procedure

  ![Diagram]

  Figure 4.461: Error Updating Peer_Clock_Accuracy

  1. The Upper Tester sends an HCI_LE_Request_Peer_SCA command to the IUT with a valid Connection_Handle.
  2. The IUT sends an HCI_Command_Status event with the error code Unsupported Feature or Parameter Value (0x11).

• Expected Outcome
  Pass Verdict
  - In step 2, the IUT sends error code Unsupported Feature or Parameter Value (0x11).
4.11 BIS

Tests that the IUT behaves according to the Broadcast Isochronous Stream (BIS) setup and BIS procedures.

4.11.1 Common Parameters

The following table defines common default values for this section. The test case may specify different values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default_TxNumBIS</td>
<td>lesser of 5 or the value of TSPX_max_tx_bises in IXIT</td>
<td>Default number of transmit BISes supported by an Isochronous Broadcaster unless otherwise specified</td>
</tr>
<tr>
<td>Default_RxNumBIS</td>
<td>lesser of 2 or the value of TSPX_max_rx_bises in IXIT</td>
<td>Default number of receive BISes supported by a Synchronized Receiver unless otherwise specified</td>
</tr>
<tr>
<td>NumDataPDUs</td>
<td>20</td>
<td>Number of ISO Data PDUs to send and/or receive as part of a test procedure</td>
</tr>
<tr>
<td>Default_RTN</td>
<td>2</td>
<td>Default explicit or implicit value of RTN</td>
</tr>
<tr>
<td>Default_NSE</td>
<td>4</td>
<td>Default number of subevents when explicitly set</td>
</tr>
<tr>
<td>Default_IRC</td>
<td>2</td>
<td>Default Immediate Repetition Count</td>
</tr>
<tr>
<td>Default_PTO</td>
<td>0</td>
<td>Default Pretransmission Offset</td>
</tr>
<tr>
<td>Default_BN</td>
<td>2</td>
<td>Default Burst Number when explicitly set</td>
</tr>
<tr>
<td>Default_Transport_Latency</td>
<td>20 ms</td>
<td>Default transport latency</td>
</tr>
<tr>
<td>Default_SDU_Interval</td>
<td>10 ms</td>
<td>Default SDU Interval</td>
</tr>
<tr>
<td>Default_ISO_Interval</td>
<td>10 ms</td>
<td>Default ISO Interval</td>
</tr>
<tr>
<td>Default_BIG_Sync_Timeout</td>
<td>100 ms</td>
<td>Default BIG Sync timeout</td>
</tr>
<tr>
<td>Default_Data_Size</td>
<td>32</td>
<td>Default data size of a BIS packet unless otherwise specified</td>
</tr>
<tr>
<td>Default_Data</td>
<td>any</td>
<td>When not otherwise specified, each data octet sent as data can have any value.</td>
</tr>
<tr>
<td>Default_PHY</td>
<td>0x01</td>
<td>LE 1M PHY</td>
</tr>
<tr>
<td>Default_Packing</td>
<td>0x00</td>
<td>Sequential</td>
</tr>
<tr>
<td>Default_Framing</td>
<td>0x00</td>
<td>Unframed</td>
</tr>
<tr>
<td>Default_Encryption</td>
<td>0x00</td>
<td>Unencrypted</td>
</tr>
</tbody>
</table>
### Configuring the ISO Data Path for HCI

The HCI_LE_Setup_ISO_Data_Path shall always be configured to use the HCI for input and output. Input_Data_Path shall be set to 0x00 (HCI transport), and likewise, Output_Data_Path shall be set 0x00 (HCI transport).

### Timing Requirements

When verifying the timing requirements, the following requirements must be met:

1. When the Link layer uses the active clock accuracy during a broadcast isochronous event to transmit a packet, even if they are in different BISes or in different subevents, the instantaneous timings shall not deviate more than 2 μs from the average timing. That is, the start of a packet shall be transmitted 150±2 μs after the end of the previous packet.

2. When the Link layer uses the sleep clock accuracy to transmit, the instantaneous timing of the anchor point shall not deviate more than 16 μs from the average timing. That is, the tolerance allowed on either side of the anchor point is sleep clock accuracy plus 16 μs.

3. The clock used to time packet reception may have any accuracy, but the receiving device will need to allow for this, and adjust the time when it starts and continues listening for packets accordingly.

### Common PDU Contents

The packet descriptions for BIS PDUs sent and accepted by the tester are displayed below. The addresses used in tests vary for the tester. The data used in tests varies.

#### AUX_SYNC_IND PDU ACAD Parameters (BiGInfo with Encryption Mode Enabled)

<table>
<thead>
<tr>
<th>BiGInfo</th>
<th>LSB</th>
<th>BiG_Offset</th>
<th>BiG_Offset_Units</th>
<th>ISO_Interval</th>
<th>Num_BIS</th>
<th>NSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiGInfo (continued)</td>
<td></td>
<td>BN</td>
<td>Sub_Interval</td>
<td>PTO</td>
<td>BISSpacing</td>
<td>IRC</td>
</tr>
<tr>
<td></td>
<td>(3 bits)</td>
<td>(20 bits)</td>
<td>(4 bits)</td>
<td>(20 bits)</td>
<td>(4 bits)</td>
<td></td>
</tr>
</tbody>
</table>
### BIGInfo (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max_PDU</td>
<td>Max_PDU</td>
<td>(8 bits)</td>
</tr>
<tr>
<td>RFU</td>
<td>RFU</td>
<td>(8 bits)</td>
</tr>
<tr>
<td>SeedAccessAddress</td>
<td>SeedAccessAddress</td>
<td>(32 bits)</td>
</tr>
<tr>
<td>SDU_Interval</td>
<td>SDU_Interval</td>
<td>(20 bits)</td>
</tr>
<tr>
<td>Max_SDU</td>
<td>Max_SDU</td>
<td>(12 bits)</td>
</tr>
</tbody>
</table>

### BIGInfo (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseCRCInit</td>
<td>BaseCRCInit</td>
<td>(16 bits)</td>
</tr>
<tr>
<td>ChM</td>
<td>ChM</td>
<td>(37 bits)</td>
</tr>
<tr>
<td>PHY</td>
<td>PHY</td>
<td>(3 bits)</td>
</tr>
<tr>
<td>bisPayloadCount</td>
<td>bisPayloadCount</td>
<td>(39 bits)</td>
</tr>
<tr>
<td>Framing</td>
<td>Framing</td>
<td>(1 bit)</td>
</tr>
</tbody>
</table>

### BIGInfo (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIV</td>
<td>GIV</td>
<td>(8 octets)</td>
</tr>
<tr>
<td>GSKD</td>
<td>GSKD</td>
<td>(16 octets)</td>
</tr>
</tbody>
</table>

### AUX_SYNC_IND PDU ACAD Parameters (BIGInfo with Encryption Mode Disabled)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG_Offset</td>
<td>BIG_Offset</td>
<td>(14 bits)</td>
</tr>
<tr>
<td>BIG_Offset_Units</td>
<td>BIG_Offset_Units</td>
<td>(1 bit)</td>
</tr>
<tr>
<td>ISO_Interval</td>
<td>ISO_Interval</td>
<td>(12 bits)</td>
</tr>
<tr>
<td>Num_BIS</td>
<td>Num_BIS</td>
<td>(5 bits)</td>
</tr>
<tr>
<td>NSE</td>
<td>NSE</td>
<td>(5 bits)</td>
</tr>
</tbody>
</table>

### BIGInfo (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>BN</td>
<td>(3 bits)</td>
</tr>
<tr>
<td>Sub_Interval</td>
<td>Sub_Interval</td>
<td>(20 bits)</td>
</tr>
<tr>
<td>PTO</td>
<td>PTO</td>
<td>(4 bits)</td>
</tr>
<tr>
<td>BIS_Spacing</td>
<td>BIS_Spacing</td>
<td>(20 bits)</td>
</tr>
<tr>
<td>IRC</td>
<td>IRC</td>
<td>(4 bits)</td>
</tr>
</tbody>
</table>
### BIGInfo (continued)

<table>
<thead>
<tr>
<th></th>
<th>Max_PDU (8 bits)</th>
<th>RFU (8 bits)</th>
<th>SeedAccessAddress (32 bits)</th>
<th>SDU_Interval (20 bits)</th>
<th>Max_SDU (12 bits)</th>
</tr>
</thead>
</table>

### MSB

<table>
<thead>
<tr>
<th></th>
<th>BaseCRCInit (16 bits)</th>
<th>ChM (37 bits)</th>
<th>PHY (3 bits)</th>
<th>bisPayloadCount (39 bits)</th>
<th>Framing (1 bit)</th>
</tr>
</thead>
</table>

### BROADCAST ISOCHRONOUS DATA PDU (Encrypted)

<table>
<thead>
<tr>
<th>Broadcast Isochronous PDU Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
</tr>
<tr>
<td>LLID '00' or '01'</td>
<td>CSSN (3 bits)</td>
</tr>
</tbody>
</table>

### BROADCAST ISOCHRONOUS DATA PDU (Unencrypted)

<table>
<thead>
<tr>
<th>Broadcast Isochronous PDU Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
</tr>
<tr>
<td>LLID '00'</td>
<td>CSSN (3 bits)</td>
</tr>
</tbody>
</table>

### BIG_CHANNEL_MAP_IND PDU

<table>
<thead>
<tr>
<th>Broadcast Isochronous PDU Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>lsb msb</td>
<td>lsb msb</td>
</tr>
<tr>
<td>LLID '11'</td>
<td>CSSN (3 bits)</td>
</tr>
</tbody>
</table>

Extra note: The table contains information about various fields and their sizes within the context of Bluetooth SIG Proprietary specifications for Link Layer (LL) test suite.
BIG_TERMINATE_IND PDU

<table>
<thead>
<tr>
<th>Broadcast Isochronous PDU Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lsb msb</td>
</tr>
<tr>
<td></td>
<td>lsb msb</td>
</tr>
<tr>
<td>LLID '11'</td>
<td>CSSN (3 bits)</td>
</tr>
<tr>
<td></td>
<td>CSTF (1 bit)</td>
</tr>
<tr>
<td></td>
<td>RFU '00'</td>
</tr>
<tr>
<td></td>
<td>Length (8 bits)</td>
</tr>
<tr>
<td></td>
<td>Reason (1 octet)</td>
</tr>
<tr>
<td></td>
<td>Instant (2 octets)</td>
</tr>
<tr>
<td>LSO MSO</td>
<td>lsb msb</td>
</tr>
</tbody>
</table>

4.11.3 BRD

Verify the behavior of the IUT acting as a BIS Isochronous Broadcaster when performing BIS setup and BIS procedures.

4.11.3.1 [Broadcast Isochronous Stream Setup]

- Test Purpose

  Test that an IUT acting as an Isochronous Broadcaster correctly sets up a BIS by sending synchronization information of the BIS using periodic advertising trains.

- Reference

  [14] Section 4.4.6

- Initial Condition

  - State: Periodic Advertising (selected primary_adv_interval_min, selected primary_adv_interval_max, supported type of own_address_type, selected periodic_adv_interval_min, selected periodic_adv_interval_max)
  - The Lower Tester is configured to monitor advertising packets from the IUT.
  - TSPX_broadcast_code is the 4–16 character Broadcast Code as defined in the IXIT [9] entry.
Test Procedure

1. The Upper Tester sends an HCI_LE_Create_BIG_Test command to the IUT to establish a single BIS and the IUT responds with an HCI_Command_Status event. If encryption is enabled, the Upper Tester shall provide a Broadcast_Code derived from TSPX_broadcast_code. Framing is 0x00 (Unframed).

2. The Upper Tester receives an HCI_LE_Create_BIG_Complete event with the NSE, BN, PTO, IRC, and Max_PDU parameters as set in step 1 from the IUT. The PHY matches the PHY used to create the BIG. NOTE: Chronologically, this step will occur after BIG begins transmission as shown in Figure 4.462.

3. The IUT inserts BIGInfo from the established BIG into the ACAD field of the extended header. BIG_Offset is equal to or greater than 600 µs. The BIG_Offset_Units bit is not set if the BIG_Offset is less than 491,460 µs. The value of BisPayloadCounter shall be set to 0 in the very first BIGInfo, and increment by one for each new Data PDU transmitted in the BIS. If specified in Table 4.137, BIGInfo includes the indicated encryption data, otherwise the encryption fields are excluded.

4. The Lower Tester receives periodic advertising packets from the IUT containing BIGInfo and it matches the criteria specified in step 3.

5. The Lower Tester receives an advertising PDU (AUX_SYNC_IND+ACAD) from the IUT and a BIS Empty Data Packet.
6. The Upper Tester sends an HCI_LE_Read_Buffer_Size [v2] command and the IUT responds with an HCI_Command_Complete event providing an ISO_Data_Packet_Length that matches the IXIT ISO Data Packet Length.

7. The Upper Tester sets up Isochronous data paths on the IUT by sending an HCI_LE_ISO_Setup_Data_Path command to the IUT.

8. The Upper Tester begins sending HCI ISO Data Packets to the IUT. If specified in Table 4.137, the data is encrypted. The data consists of alternating values of 0x6C and 0x74. The data size shall be the lesser of Default_Data_Size and the maximum buffer size as previously read from the IUT.

9. The Lower Tester receives BIS Data Packets from the IUT containing alternating values of 0x6C and 0x74. The BIS shall meet the following criteria:
   a. There shall be a BIS PDU in every subevent.
   b. The value of the Sub_Interval for a single BIS shall be greater than or equal to (Tx time of the Max_PDU of BIS Data PDU + T_MSS) and less than ISO_Interval.
   c. All subsequent subevents after the first subevent in every ISO_Interval shall use a physical channel using the subevent mapping part of the Channel Selection Algorithm #2.
   d. The BIS event closes at least T_IFS before the anchor point of the next BIS event.
   e. The offset from the AUX_SYNC_IND containing BIGInfo and the BIS anchor point matches BIG Offset.
   f. The Constant Tone Extension field is not to be present in any packets.

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Encryption</th>
<th>BIGInfo Encryption Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.11.3.1.1 LL/BIS/BRD/BV-01-C</td>
<td>Disabled</td>
<td>No encryption fields</td>
</tr>
<tr>
<td>4.11.3.1.2 LL/BIS/BRD/BV-02-C</td>
<td>Enabled</td>
<td>Encryption fields (GSKD, GIV, bisPacketCounter) Reference [14] Section 4.4.6.10</td>
</tr>
</tbody>
</table>

Table 4.137: Broadcast Isochronous Stream Setup

- **Pass Verdict**
  - The IUT sends synchronization information for the BIS using periodic advertising PDUs (AUX_SYNC_IND). The BIGInfo is received with the fields as indicated.
  - In step 2, the Upper Tester receives an HCI_LE_Create_BIG_Complete event from the IUT. The PHY field matches the PHY used to create the BIG.
  - In step 4, BIGInfo matches the specified criteria.
  - In step 7, the IUT responds with error code Unknown Connection Identifier (0x02).
  - The IUT provides an ISO_Data_Packet_Length to the Upper Tester that matches the IXIT ISO Data Packet Length.
  - BIS data is received by the Lower Tester and contains the expected data and meets all the specified criteria.
4.11.3.2  LL/BIS/BRD/BV-04-C [Data Transmission in Multiple Broadcast Isochronous Streams, Interleaved BIG]

- **Test Purpose**
  Test that an IUT acting as a Connectionless Broadcaster correctly sends data packets on multiple BISes over a Broadcast Isochronous Group in an interleaved arrangement.

- **Reference**
  [14] Section 4.4.6, 4.4.6.4, 4.4.6.5

- **Initial Condition**
  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.
  TSPX_max_tx_bises is the maximum number of supported transmit BIS by an Isochronous Broadcaster as defined in the IXIT [9] entry.

  State: Isochronous Broadcasting, Test (values as specified in Table 4.138)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_bis</td>
<td>Table 4.139 Num_BIS</td>
</tr>
<tr>
<td>sdu_int</td>
<td>Table 4.139 ISO_Interval</td>
</tr>
<tr>
<td>iso_int</td>
<td>Table 4.139 ISO_Interval</td>
</tr>
<tr>
<td>nse</td>
<td>default</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>Table 4.139 Payload Size</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>Table 4.139 Payload Size</td>
</tr>
<tr>
<td>phy</td>
<td>LE 1M PHY, then as specified</td>
</tr>
<tr>
<td>packing</td>
<td>0x01 (Interleaved)</td>
</tr>
<tr>
<td>framing</td>
<td>0x00 (Unframed)</td>
</tr>
<tr>
<td>bn</td>
<td>default</td>
</tr>
<tr>
<td>irc</td>
<td>default</td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
</tr>
<tr>
<td>encryption</td>
<td>0x00</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>n/a</td>
</tr>
</tbody>
</table>

*Table 4.138: State Variable Values*
• Test Procedure

1. The Upper Tester continuously sends HCI ISO Data Packets. Each data packet shall be filled with the BIS_Number of its respective BIS. The payload size is specified in Table 4.139 for each iteration.

2. The IUT transmits Broadcast Isochronous Data PDUs over multiple BISes in an interleaved arrangement. The Data PDUs are transmitted in compliance with the default specified BN and IRC values.

3. The Lower Tester receives data packets from the IUT in an interleaved arrangement. It receives a minimum of NumDataPDUs ISO Data Packets for each BIS. The Lower Tester confirms that the value of the CSSN in the header of every BIS PDU in a BIG event is the same. The CSTF bit is 0 in the header of every BIS Data PDU sent in that BIG event.

4. The IUT terminates the BIG.

5. Repeat steps 1–4 with each row of values from Table 4.139.

6. Repeat steps 1–5 for each PHY supported by the IUT.
### Expected Outcome

**Pass Verdict**

- The IUT transmits data packets in the correct order on multiple BISes in an interleaved arrangement. The Data PDUs are transmitted in compliance with the specified BN and IRC values and expected timing. The data matches the expected pattern.
- For all BIG events before the first one containing a BIG_TERMINATE_IND, all the PDUs have CSSN=N (where N can be any value between 0 and 7) and CSTF=0.
- For all BIG events containing a BIG_TERMINATE_IND, all the PDUs have CSSN=N+1. The data PDUs have CSTF=1 while the BIG_TERMINATE_IND has CSTF=0.
- For any BIG events after the first BIG_TERMINATE_IND but not containing one, all the PDUs have CSSN=N+1 and CSTF=0.
- The IUT ceases broadcasting isochronous data at the instant after being commanded to terminate the BIG by the Upper Tester.

#### 4.11.3.3 LL/BIS/BRD/BV-05-C [Data Transmission in Multiple Broadcast Isochronous Streams, Sequential]

- **Test Purpose**
  Test that an IUT acting as a Connectionless Broadcaster correctly sends data packets on multiple BISes in a sequential arrangement.

- **Reference**
  [14] Section 4.4.6, 4.4.6.4, 4.4.6.5

- **Initial Condition**

  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.

  State:Isochronous Broadcasting, Test (values as specified in Table 4.140)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_bis</td>
<td>Table 4.139 Num_BIS</td>
</tr>
<tr>
<td>sdu_int</td>
<td>Table 4.139 ISO_Interv</td>
</tr>
<tr>
<td>iso_int</td>
<td>Table 4.139 ISO_Interv</td>
</tr>
<tr>
<td>nse</td>
<td>default</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>Table 4.139 Payload Size</td>
</tr>
</tbody>
</table>

---

### Table 4.139: Num_BIS Values

<table>
<thead>
<tr>
<th>Round</th>
<th>Num_BIS</th>
<th>Payload Size</th>
<th>ISO_Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>24</td>
<td>40 milliseconds</td>
</tr>
<tr>
<td>2</td>
<td>Default_TxNumBIS</td>
<td>16</td>
<td>70 milliseconds</td>
</tr>
<tr>
<td>3</td>
<td>TSPX_max_tx_bises</td>
<td>8</td>
<td>250 milliseconds</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mx_pdu</td>
<td>Table 4.139 Payload Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>phy</td>
<td>LE 1M PHY, then as specified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>framing</td>
<td>0x00 (Unframed)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bn</td>
<td>default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>irc</td>
<td>default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>encryption</td>
<td>0x00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>broadcast_code</td>
<td>n/a</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.140: State Variable Values*
• Test Procedure

1. The Upper Tester continuously sends HCI ISO Data Packets. Each data packet shall be filled with the BIS_Number of its respective BIS. The payload size is specified in Table 4.139 for each iteration.

2. The IUT transmits Broadcast Isochronous Data PDUs in a sequential arrangement. The Data PDUs are transmitted in compliance with the default specified BN and IRC values.

3. The Lower Tester receives data packets from the IUT in a sequential arrangement. It receives a minimum of NumDataPDUs ISO Data Packets for each BIS. The Lower Tester confirms that the value of the CSSN in the header of every BIS Data PDU in a BIG event is the same. The CSTF bit is 0 in the header of every BIS PDU sent in that BIG event.

4. The IUT terminates the BIG.

5. Repeat steps 1–4 with each row of values from Table 4.139.

6. Repeat steps 1–5 for each PHY value supported by the IUT.

• Expected Outcome

Pass Verdict
- The IUT transmits data packets in the correct order on multiple BISes in a sequential arrangement. The Data PDUs are transmitted in compliance with the specified BN and IRC values and expected timing. The data matches the expected pattern.
- For all BIG events before the first one containing a BIG_TERMINATE_IND, all the PDUs have CSSN=N (where N can be any value between 0 and 7) and CSTF=0.
- For all BIG events containing a BIG_TERMINATE_IND, all the PDUs have CSSN=N+1. The data PDUs have CSTF=1 while the BIG_TERMINATE_IND has CSTF=0.
- For any BIG events after the first BIG_TERMINATE_IND but not containing one, all the PDUs have CSSN=N+1 and CSTF=0.
- The IUT ceases broadcasting isochronous data when commanded to terminate the BIG.

4.11.3.4 LL/BIS/BRD/BV-07-C [Bursting of Packets in Broadcast Isochronous Stream]

- **Test Procedure**
  Test that an IUT acting as a Connectionless Broadcaster sends data packets in a burst when Burst Number (BN) is greater than 1.

- **Reference**
  [14] Section 4.4.6.6

- **Initial Condition**
  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.
  State: Isochronous Broadcasting, Test (values as specified in Table 4.142)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_bis</td>
<td>Default_TxNumBIS</td>
</tr>
<tr>
<td>sdu_int</td>
<td>default</td>
</tr>
<tr>
<td>iso_int</td>
<td>default</td>
</tr>
<tr>
<td>nse</td>
<td>Table 4.142 NSE</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>default</td>
</tr>
<tr>
<td>phy</td>
<td>LE 1M PHY, then as specified</td>
</tr>
<tr>
<td>packing</td>
<td>0x00 (Sequential)</td>
</tr>
<tr>
<td>framing</td>
<td>0x00 (Unframed)</td>
</tr>
<tr>
<td>bn</td>
<td>Table 4.142 Burst Number (BN)</td>
</tr>
<tr>
<td>irc</td>
<td>default</td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
</tr>
<tr>
<td>encryption</td>
<td>0x00</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>n/a</td>
</tr>
</tbody>
</table>

*Table 4.141: State Variable Values*
Test Procedure

1. The Lower Tester synchronizes with the IUT and begins listening on multiple BISes.
2. The Upper Tester submits HCI ISO data packets to the IUT to be distributed on multiple BISes in a sequential arrangement.
3. After sending each Isochronous Data PDU, the Upper Tester sends an HCI_LE_Read_ISO_TX_Sync command. The Upper Tester receives an HCI_Command_Complete event that includes the Time_Stamp. The Upper Tester expects the value of Time_Stamp to be 0 on every read.
4. The Lower Tester receives data packets from the IUT on multiple BISes in a sequential arrangement, such that every BIS subevent data is broadcast exactly once before being repeated within an ISO_Interval. The Lower Tester receives a minimum of NumDataPDUs data packets.
5. The Upper Tester sends an HCI_LE_Terminate_BIG command to the IUT and receives an HCI_Command_Status event in response.
6. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.
7. Repeat steps 1–6 with the values for BIS_Parameters fields in Table 4.142.

<table>
<thead>
<tr>
<th>Round</th>
<th>Burst Number (BN)</th>
<th>NSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.142: Values for BN and NSE
• Expected Outcome
  Pass Verdict
  - The IUT sends the correct number of new data packets for each BIS interval.
  - In step 4, the value of Time_Stamp is 0 on every read.

4.11.3.5  LL/BIS/BRD/BV-08-C [Pre-transmissions in Broadcast Isochronous Stream]

• Test Purpose
  Test that an IUT acting as a Connectionless Broadcaster correctly retransmits data packets in a Broadcast Isochronous Stream using the pre-transmission schemes according to the PTO and IRC parameters.

• Reference
  [14] Section 4.4.6.6

• Initial Condition
  - State: Periodic Advertising (selected primary_adv_interval_min, selected primary_adv_interval_max, supported type of own_address_type, selected periodic_adv_interval_min, selected periodic_adv_interval_max)
  - TSPX_max_sdu_length is the maximum size of an SDU packet as defined in the IXIT [9] entry.
  - TSPX_max_tx_pto is the maximum supported pre-transmission offset defined in the IXIT [9] entry.

• Test Procedure

![Diagram](Image)

*Figure 4.466: LL/BIS/BRD/BV-08-C [Pre-transmissions in Broadcast Isochronous Stream]*
If TSPX_max_tx_pto is 0, refer to the Inconclusive Verdict.

1. Establish a single Broadcast ISO Stream using the established periodic advertising handle. Set the BN, IRC, PTO, NSE, and ISO_Interval values as specified in Table 4.143. The PTO value used is the lesser of TSPX_max_tx_pto and the PTO value from the table. Configure the BIS for framed PDUs.
2. The Lower Tester begins listening for advertising from the IUT (AUX_SYNC_IND+ACAD PDUs).
3. The Lower Tester performs the synchronization procedure with the IUT and successfully synchronizes with the BIS.
4. The Upper Tester continuously sends HCI ISO data packets to the IUT. The Upper Tester shall number the outgoing buffers, starting at 0x00 and wrapping back to 0x00 after 0xFF is reached. This value shall fill every octet of the outgoing data buffer. NOTE: This aids in confirming that the pre-transmission packets are offset as specified.
5. The Lower Tester receives Data PDUs from the IUT according to the following calculations:
   a. The scheduled Data PDU will be transmitted IRC times.
   b. BIS interval spacing is determined by the PTO value (e.g., if the current event is X, then the Data PDU transmitted in the first subevent in the Pre-transmission part will be the Data PDU from the event (X + (1*PTO), the second PDU from the event (X + (2*PTO), etc.).
   c. The number of subevents in the Immediate part of the transmission is equal to BN * IRC.
   d. The number of subevents in the Pre-transmission part of the transmission is equal to NSE – (BN * IRC).
   e. PTO determines interval spacing for Pre-transmission PDUs.
6. The Lower Tester receives a minimum of NumDataPDUs before continuing.
7. After sending each Isochronous Data PDU, the Upper Tester sends an HCI_LE_Read_ISO_TX_Sync command. The Upper Tester receives an HCI_Command_Complete event that includes the Packet_Sequence_Number and a Time_Stamp. The Packet_Sequence_Number shall increment by 1 for each Isochronous Data PDU sent, and the Time_Stamp shall advance by an amount that is expected and within acceptable limits.
8. The Upper Tester sends an HCI_LE_Terminate_BIG command to the IUT and receives an HCI_Command_Status event in response.
9. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.
10. Repeat steps 1–9 with the values in Table 4.143.

<table>
<thead>
<tr>
<th>Round</th>
<th>BN</th>
<th>IRC</th>
<th>PTO</th>
<th>NSE</th>
<th>ISO_Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5 ms</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>8.75 ms</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>7.5 ms</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>10 ms</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>20 ms</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>11.25 ms</td>
</tr>
</tbody>
</table>

Table 4.143: BN, IRC, PTO, and NSE Values
• Expected Outcome

Pass Verdict
- The IUT sends Broadcast Isochronous Data PDUs in the BIS according to the pre-transmission scheme.
- In steps 5 and 6, the Lower Tester receives a minimum of NumDataPDUs Data PDUs from the IUT according to the items listed in step 6.
- In step 7, the Upper Tester receives an HCI_Command_Complete event that includes the Packet_Sequence_Number and a Time_Stamp. The Packet_Sequence_Number increments by 1 for each Isochronous Data PDU sent, and the Time_Stamp advances by an amount that is expected and within acceptable limits.
- In steps 8 and 9, the BIS is successfully terminated.

Inconclusive Verdict
- TSPX_max_tx_pto is 0.

4.11.3.6 LL/BIS/BRD/BV-09-C [Broadcast Isochronous Group Channel Map Update Procedure]

• Test Purpose
  Test that an IUT acting as a Connectionless Broadcaster correctly updates the channel map for the BIG.

• Reference
  [14] Section 5.6.1

• Initial Condition
  State: Isochronous Broadcasting, Test (num_bis: 1, encryption: 0, broadcast_code: n/a, all other values as specified in Section 4.11.1 Common Parameters)
**Test Procedure**

1. The Upper Tester continuously sends HCI ISO Data Packets per the default settings specified in Section 4.11.1 Common Parameters.
2. The Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command with the Channel_Map parameter set to 0x17F7F7F7F7 to the IUT and receives an HCI_Command_Complete event in response.
3. The Lower Tester receives the BIG_CHANNEL_MAP_IND PDU with an instant in the Control subevent from the IUT.
4. The Lower Tester receives six consecutive BIG_CHANNEL_MAP_IND PDUs before the channel map update. Additional BIG_CHANNEL_MAP_IND PDUs may be sent after the six consecutive PDUs and prior to the channel map update as described in [14] Section 5.6. The Control PDUs shall have the Length field set to 8 to match the fixed size of a BIG_CHANNEL_MAP_IND Control PDU. The Instant is greater than or equal to currEventCount + 6 (currEventCount is the value of the BIS event counter when the transmission of the BIG Control PDU takes place for the first time). The value of CSSN increments by 1 with each new Control PDU (not a retransmission of a previous Control PDU) and the value of CSSN is the same in every BIS PDU in the same event. The CSTF bit is 1 in the header of every BIS Data PDU sent in that same BIG event as the Control PDU, and CSTF is set to 0 in every BIG Control PDU.
5. The Lower Tester expects the ChM field in the BIGInfo data in the associated periodic advertising chain to change to 0x17F7F7F7F7.
6. The Upper Tester commands the IUT to terminate the BIS, and the BIS is terminated (see the "BIG Termination Procedure Optional Test Steps" procedure in Section 4.1.5.7.3 Optional Test Steps).
• Expected Outcome

**Pass Verdict**

- The IUT transmits a BIG_CHANNEL_MAP_IND PDU that matches the criteria specified in step 4.
- After sending at least six BIG Control PDUs, the IUT begins transmitting PDUs over the new channel(s) at the instant specified.
- For all BIG events before the first one containing a BIG_CHANNEL_MAP_IND, all the PDUs have CSSN=N (where N can be any value between 0 and 7) and CSTF=0.
- For all BIG events containing a BIG_CHANNEL_MAP_IND, all the PDUs have CSSN=N+1. The data PDUs have CSTF=1 while the BIG_CHANNEL_MAP_IND has CSTF=0.
- For any BIG events after the first BIG_CHANNEL_MAP_IND but not containing one, all the PDUs have CSSN=N+1 and CSTF=0.
- The ChM field in the BIGInfo data in the associated periodic advertising chain changed to 0x17F7F7F7F7.
- For all BIG events containing a BIG_TERMINATE_IND, all the PDUs have CSSN=N+2. The data PDUs have CSTF=1 while the BIG_CHANNEL_MAP_IND has CSTF=0.

4.11.3.7 LL/BIS/BRD/BV-10-C [Isochronous Broadcaster Terminates BIS Stream]

• Test Purpose

Test that an IUT acting as an Isochronous Broadcaster correctly terminates an active BIS.

• Reference

[14] 4.4.6.7, 5.6.2

• Initial Condition

State: Isochronous Broadcasting, Test (num_bis: 1, encryption: 0, broadcast_code: n/a, all other values as specified in Section 4.11.1 Common Parameters)
• Test Procedure

1. The Upper Tester sends an HCI_LE_Terminate_BIG command to the IUT and receives an HCI_Command_Status event in response.

2. The IUT transmits six consecutive BIG_TERMINATE_IND PDUs in the Control subevents. Additional BIG_TERMINATE_IND commands may be sent after the six consecutive commands and prior to termination of the BIG as described in [14] Section 5.6. The following conditions shall be met:
   a. The Reason field shall be set to a valid Error Code to inform the Lower Tester why the BIG is about to be terminated.
   b. The Instant field shall be set to indicate the instant when the BIG will be terminated.
   c. The access address of the Broadcaster Control logical link shall be derived as defined in [14] Section 2.1.2.
   d. The Control subevent is scheduled BIG_Control_Offset from the BIG anchor point.
   e. The Control subevent shall use the channel index derived using the bigEventCounter in the Channel Selection algorithm #2.
   f. The value of the bisPayloadCounter for a BIG Control PDU shall be equal to the value of the bisPayloadCounter in the first subevent in the same ISO_Interval.

3. The Lower Tester observes that the termination of the BIG occurs at the Instant specified in the Control Subevent.

4. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event. The Reason parameter shall be set to the value Connection Terminated by Local Host (0x16).

5. After at least six BIS events from the IUT, the IUT sends an AUX_SYNC_IND PDU that does not contain an ACAD field containing BIGInfo to the Lower Tester.
• Expected Outcome

Pass Verdict
- The IUT transmits a BIG_TERMINATE_IND PDU.
- After the IUT transmits the BIG_TERMINATE_IND PDU sequence, the IUT sends an AUX_SYNC_IND PDU that does not contain BigInfo. The Control PDU is correctly formatted and meets the requirements specified in step 2.
- The IUT reports the HCI_LE_Terminate_BIG_Complete event with a Reason of 0x00. The Reason parameter is set to the value Connection Terminated by Local Host (0x16).

4.11.3.8 Broadcast Isochronous Stream Using Non-Test Command

• Test Purpose
Verify the IUT correctly executes the LE Create BIG Command.

• Reference
[15] Section 7.8.103

• Initial Condition
State: Periodic Advertising (selected primary_adv_interval_min, selected primary_adv_interval_max, supported type of own_address_type, selected periodic_adv_interval_min, selected periodic_adv_interval_max)
Figure 4.469: [Broadcast Isochronous Stream Using Non-Test Command] – Part A
CLEAR 2ND ADV TRAIN USING PERIODIC ADVERTISING PREAMBLE

Advertising Packet

HCI_LE_Create_BIG
(Previous_BIG_Handle)

HCI_Command_Status
(Status: 0x0C)

HCI_LE_Terminate_BIG

6 Consecutive

BIG_TERMINATE_IND PDU
Control Subevent

HCI_Command_Status

BIG_TERMINATE_IND PDU
Control Subevent

HCI_LE_Terminate_BIG_Complete

6 Consecutive

BIG_TERMINATE_IND PDU
Control Subevent

HCI_LE_Terminate_BIG_Complete

Figure 4.470: [Broadcast Isochronous Stream Using Non-Test Command] – Part B
1. The Upper Tester sends an HCI_LE_Create_BIG command using the correct Advertising_Handle obtained previously and the Max_Transport_Latency as specified in the Test Case Configurations table. Only a single PHY bit shall be set as specified in the Test Case Configurations table. The frame bit shall be set to 0b0 and encryption shall be disabled. The Upper Tester receives an HCI_Command_Status event in return.

2. The Upper Tester receives an HCI_LE_Create_BIG_Complete event from the IUT. The PHY matches the PHY used to create the BIG.

3. The Lower Tester receives advertising PDUs (AUX_SYNC_IND+ACAD) from the IUT and BIS Empty Data Packets on the PHY specified previously. BN shall not be 0. The value of NSE shall be a multiple of BN for a BIS, and shall not be 0. The IRC value shall not be set to 0. The ISO_Interval shall be a multiple of 1.25 milliseconds in the range of 5 milliseconds to 4 seconds. Max_PDU is between 0 and 251.

4. The Upper Tester sends an HCI_LE_Read_Buffer_Size [v2] command and the IUT responds with an HCI_Command_Complete event providing an ISO_Data_Packet_Length.

5. If the IUT responds with an ISO_Data_Packet_Length of 0x00, indicating that it does not support dedicated ISO transmit buffer(s), then the Upper Tester shall send an HCI_Read_Buffer_Size command to determine the length of the transmit buffer(s).

6. The Upper Tester sets up Isochronous data paths on the IUT by sending an HCI_LE_ISO_Setup_Data_Path command to the IUT.

7. The Upper Tester begins sending HCI ISO Data Packets to the IUT. The data consists of alternating values of 0x81 and 0x16. The data size shall be the lesser of Default_Data_Size and the maximum buffer size as previously read from the IUT.

8. The Lower Tester receives BIS Data Packets from the IUT containing alternating values of 0x81 and 0x16. The data packets shall be unframed.

9. The Upper Tester commands the IUT to open a second periodic advertising train per the periodic advertising preamble.

10. The Upper Tester sends an HCI_LE_Terminate_BIG command using the BIG_Handle of the existing BIG to the IUT and receives an HCI_Command_Status event in response.

11. The IUT transmits six consecutive BIG_TERMINATE_IND PDUs in the Control subevents. Additional BIG_TERMINATE_IND PDUs may be sent after the six consecutive PDUs and prior to termination of the BIG as described in [14] Section 5.6.

12. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.

13. The Upper Tester sends an HCI_LE_Create_BIG command using the Advertising_Handle obtained for the second periodic advertising train and the Max_Transport_Latency as specified in the Test Case Configurations table. Only a single PHY bit shall be set as specified in the Test Case Configurations table. The frame bit shall be set to 0b1 and encryption shall be disabled. The Upper Tester receives a successful HCI_Command_Status in return.

14. The Upper Tester sets up Isochronous data paths on the IUT by sending an HCI_LE_ISO_Setup_Data_Path command to the IUT.

15. The Upper Tester begins sending HCI ISO Data Packets to the IUT. The data consists of alternating values of 0xFA and 0xED. The data size shall be the lesser of Default_Data_Size and the maximum buffer size as previously read from the IUT.

16. The Lower Tester receives BIS Data Packets from the IUT containing alternating values of 0xFA and 0xED on the PHY specified previously. The data packets shall be framed.

17. The Upper Tester sends an HCI_LE_Terminate_BIG command using the BIG_Handle of the existing BIG to the IUT and receives an HCI_Command_Status event in response.

18. The IUT transmits six consecutive BIG_TERMINATE_IND PDUs in the Control subevents. Additional BIG_TERMINATE_IND PDUs may be sent after the six consecutive PDUs and prior to termination of the BIG as described in [14] Section 5.6.

19. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.
- **Test Case Configurations**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>PHY</th>
<th>Max_Transport_Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.11.3.8.1 LL/BIS/BRD/BV-13-C</td>
<td>LE 1M PHY</td>
<td>Default_Transport_Latency</td>
</tr>
<tr>
<td>4.11.3.8.2 LL/BIS/BRD/BV-14-C</td>
<td>LE 2M PHY</td>
<td>Default_Transport_Latency</td>
</tr>
<tr>
<td>4.11.3.8.3 LL/BIS/BRD/BV-15-C</td>
<td>LE Coded PHY</td>
<td>Default_Transport_Latency X 8</td>
</tr>
</tbody>
</table>

*Table 4.144: Test Case Configurations*

**Pass Verdict**

- In step 2, the Upper Tester receives an HCI_LE_Create_BIG_Complete event from the IUT. The PHY field matches the PHY used to create the BIG.

- In step 3, the IUT broadcasts BIS Empty Data Packets on the PHY specified in the Test Case Configurations table. The parameters match the criteria specified.

- In step 8, the IUT transmits BIS Data Packets containing alternating values of 0x81 and 0x16. The data packets are unframed.

- In step 11, the Lower Tester receives six consecutive BIG_TERMINATE_IND PDUs in Control subevents.

- In step 16, the IUT transmits BIS Data Packets containing alternating values of 0xFA and 0xED on the PHY specified in the Test Case Configurations table. The data packets are framed.

- In step 17, the Upper Tester receives an HCI_Command_Status event in response.

- In step 18, the Lower Tester receives six consecutive BIG_TERMINATE_IND PDUs in Control subevents.

- In step 19, the Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.

- The calculated transport latencies of all the BISes as defined in [17] Section 3.2.1 is less than or equal to the Max_Transport_Latency value in Table 4.144.

**4.11.3.9 LL/BIS/BRD/BV-16-C [Maximum Supported Broadcast Isochronous Groups]**

- **Test Purpose**
  Verify the IUT correctly generates its maximum supported Broadcast Isochronous Groups.

- **Reference**
  [14] Section 4.4.6.2

- **Initial Condition**
  TSPX_broadcast_code is the 4–16 character Broadcast Code as defined in the IXIT [9] entry.

  Establish TSPX_max_tx_big (defined in IXIT) in the state described as follows:

  **Note:** "default" refers to values specified in Section 4.11.1 Common Parameters.

  State: Isochronous Broadcasting, Test (values as specified in Table 4.145)
<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_bis</td>
<td>1</td>
</tr>
<tr>
<td>sdu_int</td>
<td>101.25 ms</td>
</tr>
<tr>
<td>iso_int</td>
<td>101.25 ms</td>
</tr>
<tr>
<td>nse</td>
<td>2</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>16</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>16</td>
</tr>
<tr>
<td>phy</td>
<td>default</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>bn</td>
<td>default</td>
</tr>
<tr>
<td>irc</td>
<td>0</td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
</tr>
<tr>
<td>encryption</td>
<td>0x01 if encryption is supported, otherwise 0x00</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>TSPX_broadcast_code</td>
</tr>
</tbody>
</table>

*Table 4.145: State Variable Values*

Upper Tester numbers the BIGs as they are created starting at 1.
- **Test Procedure**

![Diagram](image-url)

**Figure 4.471: LL/BIS/BRD/BV-16-C [Maximum Supported Broadcast Isochronous Groups]**

1. The Upper Tester continuously sends HCI ISO Data Packets for all of the created BISs. The data for each BIS shall consist of the numbering of the associated BIS repeated in every octet.
2. The Lower Tester receives AUX_SYNC_IND PDUs containing BIGInfo for each respective BIG. For all BIGs, the value of bits SAA15-0 in SeedAccessAddress shall differ in at least two bits.
3. The Lower Tester synchronizes to all BISes. All Access Addresses meet the requirements specified in [14] Section 2.1.2. The data content is as previously specified. For encrypted broadcasts, every GIV and GSKD for each BIG is unique from each other.
4. The Upper Tester sends an HCI_LE_Terminate_BIG command to the IUT for the first BIG created and receives an HCI_Command_Status event in response.
5. The Upper Tester immediately sends another HCI_LE_Terminate_BIG command to the IUT for the same BIG and receives an error response.
6. The IUT transmits six consecutive BIG_TERMINATE_IND PDUs in the Control subevents. Additional BIG_TERMINATE_IND PDUs may be sent after the six consecutive PDUs and prior to termination of the BIG as described in [14] Section 5.6.
7. The Lower Tester observes that the termination of the BIG occurs at the Instant specified in the Control Subevent.
8. The Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.
9. The IUT terminates the remaining BIGs.

- **Expected Outcome**

  **Pass Verdict**
  - In step 2, the SeedAccessAddress bit meets the criteria specified.
  - In step 3, all Access Addresses meet the requirements specified. The data content is as specified. For encrypted broadcasts, every GIV and GSKD for each BIG is unique from each other.
  - In step 5, the IUT sends an error response.
  - In step 6, the Lower Tester receives six consecutive BIG_TERMINATE_IND PDUs in the Control subevents.
  - In step 7, the Lower Tester observes that the termination of the BIG occurs at the Instant specified in the Control Subevent.
  - In step 8, the Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.
  - In step 9, the remaining BIGs are terminated.

4.11.4 **SNC**
Tests that the IUT behaves according to the BIS setup and BIS procedures as a Synchronized Receiver.

4.11.4.1 **Broadcast Isochronous Stream Synchronization Setup**

- **Test Purpose**
  Verify that an IUT acting as a Synchronized Receiver can synchronize with a Connectionless Broadcaster’s BIS using the BIG_Info information of the BIS in periodic advertising trains.
  The Lower Tester acts as an Isochronous Broadcaster, sending BIS synchronization information in the AUX_SYNC_IND PDUs. The Upper Tester observes the IUT getting synchronized and informing the host with the LE BIG Sync Established event.

- **Reference**
  [14] Section 4.4.6

- **Initial Condition**
  TSPX_broadcast_code is the 4–16 character Broadcast Code as defined in the IXIT [9] entry.
  State: Passive Scanning (selected scan interval, selected scan window AND All White Listed (policy for scanner)).
  The Lower Tester provides a periodic advertising stream with dummy data.
1. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT.
2. The Lower Tester establishes a BIG with a single BIS and begins sending periodic advertising trains with BIGInfo in the ACAD field of the AUX_SYNC_IND PDU. If encryption is enabled, the Lower Tester shall provide a 128-bit representation of the Broadcast_Code derived from TSPX_broadcast_code. The data in the BIS shall consist of all 0xBA.
3. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command and receives an HCI_Command_Status event in response, ordering the IUT to monitor periodic advertising packets from the Lower Tester.
4. The IUT reports the reception of periodic advertising PDUs by providing an 
   HCI_LE_Periodic_Advertising_Sync_Established event to the Upper Tester. The event returns a 
   Sync_Handle as one of its parameters.

5. The IUT sends an HCI_LE_Periodic_Advertising_Report event to the Upper Tester.

6. Immediately following sending an HCI_LE_Periodic_Advertising_Report to the Upper Tester, the 
   IUT sends an HCI_LE_BIGInfo_Advertising_Report event.

7. The Lower Tester ceases transmitting dummy data in the periodic advertising.

8. The IUT sends an HCI_LE_BIGInfo_Advertising_Report event to the Upper Tester, but does not 
   send an HCI_LE_Periodic_Advertising_Report event.

9. The Upper Tester orders the IUT to synchronize to the Lower Tester's BIG by sending an 
   HCI_LE_BIG_Create_Sync command using the Sync_Handle returned in the 
   HCI_LE_Periodic_Advertising_Sync_Established event and receives an HCI_Command_Status 
   event in response.

10. The IUT shall synchronize to the BIG and the Upper Tester receives an 
    HCI_LE_BIG_Sync_Established event.

11. The Upper Tester attempts to create an ISO input data path by sending an 
    HCI_LE_Setup_Data_Path command with the input path enabled to the IUT. The IUT responds 
    with error code Command Disallowed (0x0C).

12. The Upper Tester shall send an HCI_LE_Setup_ISO_Data_Path command to the IUT and 
    receives an HCI_Command_Complete event in response.

13. The IUT begins providing isochronous data packets from the BIS to the Upper Tester, and the 
    data matches the specified value.

If encryption is enabled, the test continues as follows:

14. The Lower Tester injects authentication errors into the BIS data packets.

15. The IUT shall stop synchronization with the BIS and notify 
    the Upper Tester by sending an 
    HCI_LE_BIG_Sync_Lost event with the Reason set to Error Code Connection Terminated due to 
    MIC Failure (0x3D).

• Expected Outcome

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Encryption</th>
<th>BIGInfo Encryption Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.11.4.1.1 LL/BIS/SNC/BV-01-C</td>
<td>Disabled</td>
<td>No encryption fields</td>
</tr>
<tr>
<td>4.11.4.1.2 LL/BIS/SNC/BV-02-C</td>
<td>Enabled</td>
<td>Encryption fields (GSKD, GIV, bisPacketCounter)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reference [14] Section 4.4.6.10</td>
</tr>
</tbody>
</table>

Table 4.146: Test Case Configurations

Pass Verdict

- The IUT synchronizes with the Lower Tester and sends an HCI_LE_BIG_Sync_Established 
  event.

- In step 6, the IUT sends an HCI_LE_BIGInfo_Advertising_Report event to the Upper Tester, 
  immediately following the HCI_LE_Periodic_Advertising_Report event. All of the parameters 
  provided in the report match the values of the BIG.

- In step 8, the IUT sends an HCI_LE_BIGInfo_Advertising_Report event to the Upper Tester, but 
  does not send an HCI_LE_Periodic_Advertising_Report event. All of the parameters provided in 
  the report match the values of the BIG.
- In step 7, the IUT responds with error code Command Disallowed (0x0C).
- BIS data is received by the Upper Tester and contains the expected data.
- If encryption is enabled, the IUT sends an HCI_LE_Sync_Lost event when authentication fails and stops synchronization with the Reason set to Error Code Connection Terminated due to MIC Failure (0x3D).

4.11.4.2 LL/BIS/SNC/BV-04-C [Data Reception in Multiple Broadcast Isochronous Streams]

- **Test Purpose**
  Test that an IUT acting as a Synchronized Receiver correctly receives data packets in multiple BISes of a Broadcast Isochronous Group in an interleaved arrangement.

- **Reference**
  [14] Section 4.4.6, 4.4.6.5

- **Initial Condition**
  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.

State: Synchronized to a Broadcast Isochronous Stream (values as specified in Table 4.147)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>padv_interval</td>
<td>default</td>
</tr>
<tr>
<td>big_sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>adv_phy</td>
<td>default</td>
</tr>
<tr>
<td>num_bis</td>
<td>2</td>
</tr>
<tr>
<td>sdu_int</td>
<td>default</td>
</tr>
<tr>
<td>iso_int</td>
<td>default</td>
</tr>
<tr>
<td>nse</td>
<td>default</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>default</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>bn</td>
<td>default</td>
</tr>
<tr>
<td>irc</td>
<td>default</td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
</tr>
<tr>
<td>State Variable</td>
<td>Value(s)</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>encryption</td>
<td>0x01 if encryption supported, 0x00 otherwise</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>TSPX_broadcast_code</td>
</tr>
</tbody>
</table>

*Table 4.147: State Variable Values*
• Test Procedure

1. The Upper Tester sends an HCI_LE_Remove_ISO_Data_Path command to the IUT with an invalid Connection_Handle and expects the IUT to return an error code Unknown Connection Identifier (0x02).

Figure 4.473: LL/BIS/SNC/BV-04-C [Data Reception in Multiple Broadcast Isochronous Streams]
2. The IUT is directed to sync to the Lower Tester’s BIG with two BISes but the Upper Tester provides an incorrect Broadcast_Code in the HCI_LE_BIG_Create_Sync command.
3. The Upper Tester expects the IUT to respond to the HCI_LE_BIG_Create_Sync command with the error code Unsupported Remote Feature / Unsupported LMP Feature (0x1A).
4. The Upper Tester attempts to establish a sync with a BIG with two BISes but the Upper Tester provides a Sync_Handle that does not exist in the HCI_LE_BIG_Create_Sync command.
5. The Upper Tester expects the IUT to respond to the HCI_LE_BIG_Create_Sync command with the error code Unknown Advertising Identifier (0x42).
6. The Upper Tester attempts to establish a sync to a BIG with three BISes by sending an HCI_LE_BIG_Create_Sync command to the IUT with the Num_BIS parameter set to 3.
7. The Upper Tester expects the IUT to respond to the HCI_LE_BIG_Create_Sync command with the error code Unsupported Feature or Parameter Value (0x11).
8. The Upper Tester attempts to establish a sync to a BIG but using the opposite encryption from the BIG established by the Lower Tester. If the BIG is encrypted, the Upper Tester commands the IUT to sync to it unencrypted, and if the BIG is not encrypted, the Upper Tester attempts to connect to it encrypted.
9. The Upper Tester receives the error Encryption Mode Not Acceptable (0x25) from the IUT.
10. Synchronize the IUT to the Lower Tester’s BIG, passing the broadcast Isochronous data to the Upper Tester, but noting the substeps identified as follows:
   a. Immediately after the HCI_LE_BIG_Create_Sync command is sent by the Upper Tester to the IUT and HCI_Command_Status event is received in response, and before the HCI_LE_BIG_Sync_Established event is received by the Upper Tester, the Upper Tester sends an additional HCI_LE_BIG_Create_Sync command. The Upper Tester expects the IUT to respond to the new command with error code Command Disallowed (0x0C).
   b. Before sending the HCI_LE_Setup_ISO_Data_Path command, the Upper Tester sends the IUT an HCI_LE_Remove_ISO_Data_Path command and expects the IUT to return error code Command Disallowed (0x0C).
11. The Upper Tester sends an additional HCI_LE_BIG_Create_Sync command and attempts to synchronize to a BIG using the same associated periodic advertising train as the one that it is already synchronized to, and receives an error response from the IUT.
12. The Upper Tester sends an additional HCI_LE_BIG_Create_Sync command to the IUT using the same BIG_Handle as the established BIG.
13. The Upper Tester expects the IUT to respond to the HCI_LE_BIG_Create_Sync command with the error code Unsupported Feature or Parameter Value (0x11).
14. The Lower Tester transmits Broadcast Isochronous Data PDUs to the IUT in multiple BISes in the Broadcast Isochronous Group.
15. The Upper Tester observes the IUT presenting the data received from the Lower Tester and the content of the Isochronous data is correct.
16. The Upper Tester sends an HCI_LE_Remove_ISO_Data_Path command and observes that the ISO data is no longer reported by the IUT.

• Expected Outcome

  Pass Verdict
  - In step 1, the IUT returns error code Unknown Connection Identifier (0x02).
  - In step 3, the IUT returns error code Unsupported Remote Feature / Unsupported LMP Feature (0x1A).
  - In step 5, the IUT returns error code Unknown Advertising Identifier (0x42).
  - In step 7, the IUT returns error code Unsupported Feature or Parameter Value (0x11).
- In step 9, the Upper Tester receives the error Encryption Mode Not Acceptable (0x25) from the IUT.
- In step 10a, the IUT returns error code Command Disallowed (0x0C).
- In step 10b, the IUT returns error code Command Disallowed (0x0C).
- In step 13, the IUT returns error code Unsupported Feature or Parameter Value (0x11).
- The IUT correctly receives data packets in multiple BISes and the content of the isochronous data is correct.
- When the HCI_LE_Remove_Data_Path command is sent, BIS data is no longer sent to the Upper Tester by the IUT.

4.11.4.3 LL/BIS/SNC/BV-07-C [Bursting of Packets in Broadcast Isochronous Stream]

• Test Purpose

Test that an IUT acting as a Synchronized Receiver correctly receives data packets sent in bursts in a Broadcast Isochronous Group.

The Lower Tester acts as an Isochronous Broadcaster sending data packets in a BIS with BN > 1. The Upper Tester observes the IUT presenting the data received from the IUT to the host in the correct order.

• Reference

[14] Section 4.4.6.6

• Initial Condition

Note: “default” refers to values specified in Section 4.11.1 Common Parameters.

State: Synchronized to a Broadcast Isochronous Stream (values as specified in Table 4.148)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>padv_interval</td>
<td>default</td>
</tr>
<tr>
<td>big_sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>adv_phy</td>
<td>default</td>
</tr>
<tr>
<td>num_bis</td>
<td>1</td>
</tr>
<tr>
<td>sdu_int</td>
<td>default</td>
</tr>
<tr>
<td>iso_int</td>
<td>default</td>
</tr>
<tr>
<td>nse</td>
<td>Table 4.149 NSE</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>default</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
</tbody>
</table>
### Table 4.148: State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>bn</td>
<td>Table 4.149 Burst Number (BN)</td>
</tr>
<tr>
<td>irc</td>
<td>0x01</td>
</tr>
<tr>
<td>pto</td>
<td>default</td>
</tr>
<tr>
<td>encryption</td>
<td>0x00</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>n/a</td>
</tr>
</tbody>
</table>

- **Test Procedure**

  1. The Lower Tester sends data using the values of BN and NSE from Table 4.149. The data shall consist of alternating values of 0x54 and 0xD1.
  2. The Upper Tester expects the IUT to report the reception of data from the Lower Tester.
  3. The Lower Tester terminates the BIS.
  4. Repeat steps 1–4 for each row of values for BN and NSE in Table 4.149.

### Table 4.149: BN and NSE Values

<table>
<thead>
<tr>
<th>Round</th>
<th>Burst Number (BN)</th>
<th>NSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

*Figure 4.474: LL/BIS/SNC/BV-07-C [Bursting of Packets in Broadcast Isochronous Stream]*
• Expected Outcome

  Pass Verdict
  - The IUT receives the data from the Lower Tester and presents the data to the Upper Tester.

4.11.4.4 LL/BIS/SNC/BV-08-C [Pre-transmissions in Broadcast Isochronous Stream]

• Test Purpose

  Test that an IUT acting as a Synchronized Receiver correctly receives data packets when the pre-transmission scheme is used in a Broadcast Isochronous Group.

  The Lower Tester acts as a Connectionless Broadcaster sending data packets on a BIS with PTO > 1 and IRC > 1. The Upper Tester observes the IUT receiving the data packets send by the IUT and presenting them to the host in the correct order.

• Reference

  [14] Section 4.4.6.6

• Initial Condition

  State: Passive Scanning (selected scan interval, selected scan window AND All White Listed (policy for scanner)).

  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.

  State: Synchronized to a Broadcast Isochronous Stream (values as specified in Table 4.150)

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>padv_interval</td>
<td>default</td>
</tr>
<tr>
<td>big_sync_timeout</td>
<td>default</td>
</tr>
<tr>
<td>adv_phy</td>
<td>default</td>
</tr>
<tr>
<td>num_bis</td>
<td>1</td>
</tr>
<tr>
<td>sdu_int</td>
<td>default</td>
</tr>
<tr>
<td>iso_int</td>
<td>Table 4.143 ISO_Int</td>
</tr>
<tr>
<td>nse</td>
<td>Table 4.143 NSE</td>
</tr>
<tr>
<td>mx_sdu</td>
<td>default</td>
</tr>
<tr>
<td>mx_pdu</td>
<td>default</td>
</tr>
<tr>
<td>packing</td>
<td>default</td>
</tr>
<tr>
<td>framing</td>
<td>default</td>
</tr>
<tr>
<td>bn</td>
<td>Table 4.143 BN</td>
</tr>
<tr>
<td>irc</td>
<td>Table 4.143 IRC</td>
</tr>
</tbody>
</table>
### State Variable Values

<table>
<thead>
<tr>
<th>State Variable</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pto</td>
<td>Lesser of IXIT Max Supported RX PTO and Table 4.143 PTO</td>
</tr>
<tr>
<td>encryption</td>
<td>0x00</td>
</tr>
<tr>
<td>broadcast_code</td>
<td>n/a</td>
</tr>
</tbody>
</table>

*Table 4.150: State Variable Values*

- **Test Procedure**

![Test Procedure Diagram]

1. The Lower Tester sends data using the values from Table 4.143. The Lower Tester shall number the outgoing buffers, starting at 0x00 and wrapping back to 0x00 after 0xFF is reached. This value shall fill every octet of the outgoing data buffer. NOTE: This aids in confirming that the pre-transmission packets are offset as specified.
2. The Upper Tester expects the IUT to report the reception of data from the Lower Tester. The Upper Tester shall receive a minimum of NumDataPDUs before continuing.
3. The Lower Tester terminates the BIS.
4. The Upper Tester receives the HCI_LE_BIG_Sync_Lost event with Error Code Remote User Terminated Connection (0x13) returned in the Reason field.
5. Repeat steps 1–4 for each row of values in Table 4.143.

- **Expected Outcome**

  **Pass Verdict**

  - The IUT receives the correct data from the Lower Tester and presents it in the correct order to the Upper Tester according to the pre-transmission scheme over the BIS.
4.11.4.5 LL/BIS/SNC/BV-09-C [Broadcast Isochronous Group Channel Map Update Procedure]

- Test Purpose
  Test that an IUT acting as a Synchronized Receiver correctly handles the Channel Map Update Indication and BIG Channel Map Update procedures.

  The Lower Tester acts as an Isochronous Broadcaster, sending control information in the Control subevent.

- Reference
  [14] Section 5.6.1

- Initial Condition

  Note: “default” refers to values specified in Section 4.11.1 Common Parameters.

  State: Synchronized to a Broadcast Isochronous Stream (num_bis: 1, encryption: 0, all other values as specified in Section 4.11.1 Common Parameters)
• Test Procedure

Figure 4.476: LL/BIS/SNC/BV-09-C [Broadcast Isochronous Group Channel Map Update Procedure]

1. The Lower Tester maintains a continuous stream of periodic advertising and CIS data throughout this procedure. The periodic advertising payloads shall consist of all octets set to 0xC5 and have a length of 31 octets. The BIS payloads shall consist of all octets set to 0xBA.
2. The Lower Tester ceases extended advertising but continues periodic advertising. Steps 3–5 and steps 6–8 occur in parallel.
3. The Lower Tester begins sending AUX_SYNC_IND PDUs with an extended header that contains a Channel Map Update Indication with a channelMapNEW of 0x1939393939, for a minimum of 6 events prior to the instant.
4. At the instant, the Lower Tester begins using \textit{channelMap}^{\text{NEW}}.
5. The IUT maintains the sync with the periodic advertising after the channel map change and continues to provide the expected data to the Upper Tester.
6. The Lower Tester transmits a BIG\_CHANNEL\_MAP\_IND PDU with the Channel\_Map parameter set to 0x17F7F7F7F7 containing an Instant field indicating when the channel map update will apply. The instant shall be the closest instant following the instant the periodic advertising switches to the new channel map.
7. The Lower Tester sends at least six consecutive BIG\_CHANNEL\_MAP\_IND PDUs before transmitting over the changed channel. Additional BIG\_CHANNEL\_MAP\_IND PDUs may be sent after the six consecutive PDUs and prior to termination of the BIG as described in [14] Section 5.6. The Instant is greater than currEventCount + 6 (currEventCount is the value of the BIS event counter when the transmission of the BIG Control PDU takes place for the first time).
8. The IUT maintains the sync after the channel map change and continues to provide the expected data to the Upper Tester.

- Expected Outcome

Pass Verdict
- The IUT maintains the periodic advertising and BIS syncs, and the Upper Tester receives the expected data throughout execution of the test.

4.11.4.6 LL/BIS/SNC/BV-10-C [Broadcast Isochronous Stream Termination]

- Test Purpose

Test that an IUT acting as a Synchronized Receiver correctly handles termination of an active BIS, either from the Isochronous Broadcaster or commanded by the local host.

The Lower Tester terminates a BIS.

- Reference

[14] Section 4.4.6.2

- Initial Condition

- State: Passive Scanning (selected scan interval, selected scan window AND All White Listed (policy for scanner).

- State: Synchronized to a Broadcast Isochronous Stream (num\_bis: 1, encryption: 0x00, broadcast\_code: n/a, all other values as specified in Section 4.11.1 Common Parameters)
Test Procedure

1. The Lower Tester transmits six consecutive BIG_TERMINATE_IND PDU including an Instant field and stops transmitting the ACAD field of AUX_SYNC_IND PDUs in its periodic advertising trains for the BIS.
2. The Upper Tester receives the HCI_LE_BIG_Sync_Lost event with Error Code Remote User Terminated Connection (0x13) returned in the Reason field.
3. Synchronize the IUT with the Lower Tester.
4. The Upper Tester sends an HCI_LE_BIG_Terminate_Sync command and receives an HCI_LE_BIG_Create_Sync_Established event with the Error Code Connection Terminated By Local Host (0x16).
5. Synchronize the IUT with the Lower Tester.
6. The Upper Tester sends an additional HCI_LE_BIG_Terminate_Sync command to the IUT using the previous BIG_Handle. The Upper Tester expects the IUT to respond with Status parameter Unknown Advertising Identifier (0x42).

Expected Outcome

Pass Verdict

- In step 2, the Upper Tester receives the HCI_LE_BIG_Sync_Lost event with Error Code Remote User Terminated Connection (0x13) returned in the Reason field.
- The IUT correctly handles termination of the active BIS when initiated by the Isochronous Broadcaster, as well as when initiated by the Upper Tester.
- In step 4, the IUT generates an HCI_LE_BIG_Create_Sync_Established event with the Error Code Connection Terminated By Local Host (0x16).
- In step 6, the IUT responds with error code Unknown Advertising Identifier (0x42).

4.11.4.7 LL/BIS/SNC/BV-11-C [Loss of Sync with an Isochronous Broadcaster]

- **Test Purpose**
  Test that an IUT acting as a Synchronized Receiver correctly handles loss of sync with an Isochronous Broadcaster.

- **Reference**
  [15] Section 7.7.65.30

- **Initial Condition**
  State: Synchronized to a Broadcast Isochronous Stream (num_bis: 1, encryption: 0x00, broadcast_code: n/a, all other values as specified in Section 4.11.1 Common Parameters)

- **Test Procedure**

  ![Diagram](image)

  **Figure 4.478: LL/BIS/SNC/BV-11-C [Loss of Sync with an Isochronous Broadcaster]**

1. The Upper Tester sends an HCI_LE_Terminate_BIG command to the IUT and expects the IUT to respond with Command Disallowed (0x0C).
2. The Lower Test abruptly terminates all transmissions.
3. The Upper Tester expects the IUT to send an HCI_LE_BIG_Sync_Lost event containing the BIG_Handle of the previously established BIG and a Reason field containing the error code Connection Timeout (0x08).
• Expected Outcome
  
  **Pass Verdict**
  - In step 1, the IUT responds with Command Disallowed (0x0C).
  - The IUT sends an HCI_LE_BIG_Sync_Lost event containing the Connection_Handle of the previously established BIG and a Reason field containing the error code Connection Timeout (0x08).

### 4.11.4.8 LL/BIS/SNC/BV-12-C [Broadcast Isochronous Stream Synchronization, Number of BISes Not Supported]

- **Test Purpose**
  Verify that an IUT acting as a Synchronized Receiver returns the appropriate error when the Upper Tester requests the IUT to synchronize to a BIG with more BISes than the IUT supports.

- **Reference**
  [15] Section 7.8.106

- **Initial Condition**
  
  TSPX_max_rx_bises is the maximum number of supported receive BIS by a Synchronized Receiver as defined in the IXIT [9] entry.

  State: Passive Scanning (selected scan interval, selected scan window AND All White Listed (policy for scanner)).

- **Test Procedure**

  ![Diagram](image-url)

  *Figure 4.479: LL/BIS/SNC/BV-12-C [Broadcast Isochronous Stream Synchronization, Number of BISes Not Supported]*

  If TSPX_max_rx_bises is 31, refer to the Inconclusive Verdict.
1. The Upper Tester receives an HCI_LE_Extended_Advertising_Report event from the IUT.
2. The Lower Tester establishes a BIG with 31 BISes with a 125 ms ISO Interval and otherwise according to the values specified in Section 4.11.1 Common Parameters, and begins sending periodic advertising trains with BIGNfo in the ACAD field of AUX_SYNC_IND PDU.
3. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create.Sync command and receives an HCI_Command_Status event in response, ordering the IUT to monitor periodic advertising packets from the Lower Tester.
4. The Upper Tester expects the IUT to report the reception of periodic advertising PDUs by providing an HCI_LE_Periodic_Advertising_Sync_Established event. The event returns a Sync_Handle as one of its parameters.
5. The Upper Tester orders the IUT to synchronize to the Lower Tester’s BIG by sending an HCI_LE_BIG_Create.Sync command using the Sync_Handle returned in the HCI_LE_Periodic_Advertising_Sync_Established event.
6. The Upper Tester expects the IUT to send an HCI_Command_Status response with error code Connection Rejected Due to Limited Resources (0x0D).

• Expected Outcome

Pass Verdict
  - The Upper Tester receives error code Connection Rejected Due to Limited Resources (0x0D) in response to the HCI_LE_BIG_Create.Sync command.

Inconclusive Verdict
  - TSPX_max_rx_bises is 31.

4.12 Isochronous Testing

Tests that the IUT properly exchanges test payloads in ISO Data PDUs as specified in the Isochronous Test Mode Specifications.

4.12.1 Common Test Parameter Settings

Common variables, timing, and PDU contents as defined in the related proceeding sections are to be used in this section unless otherwise specified.

4.12.2 [ISO Transmit Test Mode, CIS]

• Test Purpose
  Verify that the IUT properly transmits test payloads in a CIS connection.

• Reference
  [14] Section 7.1

• Initial Condition
  Connected in the relevant role as defined in the following initial states:
  
  State: Connected Isochronous Stream, Master, Test (values as specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands), with the exception that the HCI_LE_Setup_ISO_Data_Path command is not executed once the CIG is established.

  State: Connected Isochronous Stream, Slave (values as specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands), with the exception that the HCI_LE_Setup_ISO_Data_Path command is not executed once the CIG is established.
• Test Procedure

For each round from 1 to 3

1. The Upper Tester sends the HCI_LE_ISO_Transmit_Test command with PayloadType as specified in Table 4.151 and receives a successful HCI_Command_Complete event from the IUT in response.
2. The IUT sends isochronous data PDUs with Payload as specified in Table 4.151. The Packet Counter value shall meet the requirements for unframed PDUs as specified in Section X.1.
3. Repeat step 2 for 5 rounds of payloads.
4. The Upper Tester sends the HCI_LE_ISO_Test_End command to the IUT and receives an HCI_Command_Status event from the IUT with the Status field set to Success.

<table>
<thead>
<tr>
<th>Round</th>
<th>Payload_Type</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Each isochronous data PDU has a payload length = 0.</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>The first four octets of the isochronous data PDU contains a single 32-bit payload counter value. The remaining data is vendor specific.</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>The first four octets of the isochronous data PDU contains a single 32-bit payload counter value. The remaining data is vendor specific.</td>
</tr>
</tbody>
</table>

Table 4.151: ISO Transmit Test Mode, CIS Rounds
**Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.12.2.1 LL/IST/MAS/BV-01-C</td>
<td>Master</td>
</tr>
<tr>
<td>4.12.2.2 LL/IST/SLA/BV-01-C</td>
<td>Slave</td>
</tr>
</tbody>
</table>

*Table 4.152: ISO Transmit Test Mode, CIS Test Case Configuration*

**Pass Verdict**
- The Upper Tester successfully starts the ISO Transmit Test in step 1.
- The IUT successfully sends five rounds of isochronous data PDUs to the Lower Tester.

### 4.12.3 LL/IST/BRD/BV-01-C [ISO Transmit Test Mode, BIS]

- **Test Purpose**
  Verify that the IUT properly transmits test payloads in a BIS connection.

- **Reference**
  [14] Section 7.1

- **Initial Condition**

  State: Periodic Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, primary, supported type of own address, selected Periodic_Adv_interval_Min, selected Periodic_Adv_Interval_Max)
• Test Procedure

1. The Upper Tester sends the HCI_LE_Create_BIG_Test command to the IUT and receives a successful HCI_Command_Complete event from the IUT in response.

For each round in Table 4.151, execute steps 2–5:

2. The Upper Tester sends the HCI_LE_ISO_Transmit_Test command with Payload_Type as specified in Table 4.151 and receives a successful HCI_Command_Complete event from the IUT in response.

3. The IUT sends isochronous data PDUs with Payload as specified in Table 4.151. The Packet Counter value shall meet the requirements for unframed PDUs as specified in [14] Section X.1.

4. Repeat step 3 for five rounds of payloads.

5. The Upper Tester sends the HCI_LE_ISO_Test_End command to the IUT and receives an HCI_Command_Status event from the IUT with the Status field set to Success. The returned Received_Packet_Count, Missed_Packet_Count, and Failed_Packet_Count are all zero.

6. The Upper Tester sends the HCI_LE_Terminate_BIG command using the assigned BIG_Handle and the Reason field set to a valid Error Code, and receives an HCI_LE_Terminate_BIG_Complete event in response.
• Expected Outcome

Pass Verdict

- The Upper Tester successfully starts the ISO Transmit Test in step 2.
- The IUT successfully sends five rounds of isochronous data PDUs to the Lower Tester.
- In step 5, The returned ReceivedPacketCounter, MissedPacketCounter, and FailedPacketCounter are all zero.
- In step 6, the Upper Tester receives an HCI_LE_Terminate_BIG_Complete event from the IUT.

4.12.4 [ISO Receive Test Mode, CIS]

• Test Purpose

Verify that the IUT properly receives ISO Receive Test PDUs in a CIS connection.

• Reference

[14] Section 7.2

• Initial Condition

Connected in the relevant role as defined in the following initial states:

State: Connected Isochronous Stream, Master, Test (values as specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands), with the exception that the HCI_LE_Setup_ISO_Data_Path command is not executed once the CIG is established.

State: Connected Isochronous Stream, Slave (values as specified in Section 4.10.1.3 Default Values for Set CIG Parameters Commands), with the exception that the HCI_LE_Setup_ISO_Data_Path command is not executed once the CIG is established.
For each round from 1 to 5
Lower Tester
Upper Tester

For each round from 1 to 5

For each round from 1 to 5

For each round from 1 to 5

For each round from 1 to 5

For each round from 1 to 5

For each round from 1 to 5

For each round from 1 to 5

Figure 4.482: [ISO Receive Test Mode, CIS]

For each round in Table 4.153, execute steps 1–7:

1. The Upper Tester sends the HCI_LE_ISO_Receive_Test command to the IUT with PayloadType as specified in Table 4.153 and receives a successful HCI_Command_Complete event from the IUT in response.
2. The Lower Tester sends an SDU as specified in the table.
3. The IUT receives the expected test Payload PDU as specified in the table.
4. Repeat steps 2 and 3 for five rounds of payloads.
5. The Upper Tester sends the HCI_LE_ISO_Read_Test_Counters command to the IUT.
6. The Upper Tester receives a return parameter of Received_packet_Count as specified in Table 4.153.
7. The Upper Tester sends the HCI_LE_ISO_Test_End command to the IUT and receives an HCI_Command_Status event from the IUT with the Status field set to Success.

<table>
<thead>
<tr>
<th>Round</th>
<th>Framing</th>
<th>Payload_Type</th>
<th>Payload</th>
<th>Received_Packet_Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Each ISO data PDU has a payload length = 0.</td>
<td>5</td>
</tr>
</tbody>
</table>
Variable length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized at 0, in LSB order.

Maximum length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized to 100, in LSB order.

Variable length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized to 100, in LSB order.

Table 4.153: ISO Receive Test Mode, CIS Rounds

<table>
<thead>
<tr>
<th>Round</th>
<th>Framing</th>
<th>Payload_Type</th>
<th>Payload</th>
<th>Received_Packet_Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>Variable length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized at 0, in LSB order.</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>The first 4 octets of each SDU shall contain a Packet Counter value initialized at 0, in LSB order.</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>Maximum length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized to 100, in LSB order.</td>
<td>105</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Variable length SDUs. The first 4 octets of each SDU shall contain a Packet Counter value initialized to 100, in LSB order.</td>
<td>105</td>
</tr>
</tbody>
</table>

Table 4.154: ISO Receive Test Mode, CIS Test Case Configuration

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.12.4.1</td>
<td>LL/IST/MAS/BV-03-C</td>
</tr>
<tr>
<td>4.12.4.2</td>
<td>LL/IST/SLA/BV-03-C</td>
</tr>
</tbody>
</table>

Pass Verdict

For all rounds described in the test procedure, the following conditions shall occur:

- The IUT successfully receives one round of isochronous data PDUs from the lower tester in step 3.
- In step 6, the Received_Packet_Count value is the same as specified in Table 4.153.

4.12.5 LL/IST/SNC/BV-01-C [ISO Receive Test Mode, BIS]

- **Test Purpose**

  Verify that the IUT properly receives ISO Receive Test PDUs in a BIS connection.

- **Reference**

  [14] Section 7.2

- **Initial Condition**

  Synchronization has been established between the IUT and Lower Tester.
• Test Procedure

![Test Procedure Diagram]

**Figure 4.483: LL/IST/SNC/BV-01-C [ISO Receive Test Mode, BIS]**

1. The Upper Tester sends the HCI_LE_BIG_Create_Sync command to the IUT and receives a successful HCI_Command_Status event from the IUT in response.
2. The Upper Tester receives a successful HCI_LE_BIG_Sync_Established event from the IUT.

For each round in Table 4.153, execute steps 3–10:

3. The Upper Tester sends the HCI_LE_ISO_Receive_Test command to the IUT with Payload_Type as specified in Table 4.153 and receives a successful HCI_Command_Complete event from the IUT in response.
4. The Lower Tester sends an SDU as specified in Table 4.153.
5. The IUT receives the expected test Payload PDU as specified in Table 4.153.
6. Repeat steps 4 and 5 for five rounds of payloads.
7. The Upper Tester sends the HCI_LE_ISO_Read_Test_Counters command to the IUT.
8. The Upper Tester receives a return parameter of Received_Packet_Count as specified in Table 4.153.

9. The Upper Tester sends the HCI_LE_ISO_Test_End command to the IUT and receives an HCI_Command_Status event from the IUT with the Status field set to Success.

10. The Upper Tester sends the HCI_LE_Terminate_BIG command using the assigned BIG_Handle and the Reason field set to a valid Error Code, and receives an HCI_LE_Terminate_BIG_Complete event in response.

**Expected Outcome**

**Pass Verdict**

For all rounds described in the test procedure, the following conditions shall occur:

- The IUT successfully receives one round of isochronous data PDUs from the Lower Tester in step 3.
- In step 6, the Received_Packet_Count value is the same as specified in Table 4.153.
- In step 8, the Upper Tester receives a return parameter of Received_Packet_Count as specified in Table 4.153.
- In step 9, the Upper Tester receives an HCI_Command_Status event from the IUT with the Status field set to Success.
- In step 10, the Upper Tester receives an HCI_LE_Terminate_BIG_Complete event.

### 4.12.6 [ISO Receive Test Mode missing PDU]

**Test Purpose**

Verify that the IUT properly increments counters when incoming test PDUs are missing or invalid.

**Reference**

[14] Section 7.2

**Initial Condition**

An ACL connection has been established between the IUT and Lower Tester with a valid Connection Handle.

The IUT acts in the role as specified in Table 4.155.
1. Establish a CIS for the appropriate IUT role as described in Section 4.12.4 [ISO Receive Test Mode, CIS].
2. The Upper Tester sends the HCI_LE_ISO_Receive_Test command to the IUT with Payload_Type=0 and receives a successful HCI_Command_Complete event from the IUT in response.
3. The Lower Tester sends an invalid test PDU where the data doesn't match Payload_Type, PayloadCounter, and Max_PDU.
4. The Lower Tester sends valid test PDUs.
5. The Lower Tester skips one test PDU during a scheduled subevent.
6. The Lower Tester sends valid test PDUs.
7. The Lower Tester skips one test PDU during a scheduled subevent.
8. The Lower Tester sends valid test PDUs.
9. The Upper Tester sends the HCI_LE_ISO_Read_Test_Counters command to the IUT.
10. The Upper Tester receives a return parameter of FailedPLCounter=1 and MissedPLCounter=2.
11. The Upper Tester sends the HCI_LE_ISO_Test_End command to the IUT and receives an HCI_Command_Status event from the IUT with the Status field set to Success.

- **Expected Outcome**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.12.6.1</td>
<td>LL/IST/MAS/BV-05-C</td>
</tr>
<tr>
<td>4.12.6.2</td>
<td>LL/IST/SLA/BV-05-C</td>
</tr>
</tbody>
</table>

*Table 4.155: ISO Receive Test Mode missing PDU Test Case Configuration*

**Pass Verdict**
- In step 2, the Upper Tester receives a successful HCI_Command_Complete event from the IUT in response.
- In step 10, the FailedPLCounter and MissedPLCounter are the expected values.
- In step 11, the Upper Tester receives an HCI_Command_Status event from the IUT with the Status field set to Success.

### 4.13 PCL

Test of the Power Control procedures.

#### 4.13.1 RF Test Conditions

Some Power Control tests require altering the RF power level received by the IUT from the Lower Tester. The Lower Tester should know the golden range (declared as IXIT). The Lower Tester shall have an accuracy not worse than +/- 3dB in transmitted power level. The test conditions can be achieved by one or more of the following means:

1. The Lower Tester and IUT are connected with an RF cable and an RF attenuator to give sufficient measurement accuracy. TSPX_RF_Attenuator, defined in IXIT, is used to give the value for the RF attenuation calculated from the IUT's golden range, the cable loss and the Lower Tester's TX power range.
2. The setup of the Lower Tester, RF equipment and/or facilities provides sufficient measurement accuracy to operate the IUT within the IUT’s golden range, as well as sufficient TX power range outside the golden range to warrant power control corrections by the IUT.
4.13.2 Common PDU Contents

4.13.2.1 Power Control PDUs

LL_POWER_CONTROL_REQ:

CtrData field:

<table>
<thead>
<tr>
<th>CtrData</th>
<th>PHY (8 bits)</th>
<th>Delta (1 octet)</th>
<th>TxPower (1 octet)</th>
</tr>
</thead>
</table>

LL_POWER_CONTROL_RSP:

CtrData field:

<table>
<thead>
<tr>
<th>CtrData</th>
<th>Min (1 bit)</th>
<th>Max (1 bit)</th>
<th>RFU (6 bits) '00'</th>
<th>Delta (1 octet)</th>
<th>TxPower (1 octet)</th>
<th>APR (1 octet)</th>
</tr>
</thead>
</table>

LL_POWER_CHANGE_IND

CtrData field:

<table>
<thead>
<tr>
<th>CtrData</th>
<th>PHY (8 bits)</th>
<th>Min (1 bit)</th>
<th>Max (1 bit)</th>
<th>RFU (6 bits) '00'</th>
<th>Delta (1 octet)</th>
<th>TxPower (1 octet)</th>
</tr>
</thead>
</table>

4.13.3 Both Connected Roles

4.13.3.1 [Path Loss Monitoring]

- Test Purpose
  Verify that the IUT performs the path loss monitoring operation on an ACL connection. The IUT notifies the Upper Tester of a path loss threshold event when path loss has entered the high/middle/low zones.

- Test Case IDs
  LL/PCL/SLA/BV-01-C
  LL/PCL/MAS/BV-01-C

- Reference
  [14] Section 4.5.16

- Initial Condition
  - Parameters: TSPX_Path_Loss_Lower_Boundary, TSPX_Path_Loss_Upper_Boundary (specified in IXIT). The Lower Tester and IUT are configured as specified in Section 4.13.1 RF Test Conditions.
The system under test is initially set such that the apparent path loss as seen by the IUT places it in the Path Loss Middle Zone.

An ACL connection has been established between the IUT and the Lower Tester in the relevant role.

- **Test Procedure**

1. The Lower Tester continuously transmits empty data packets over the ACL connection with a connection interval of 7.5 ms.
2. The Upper Tester sends an `HCI_LE_Set_Path_Loss_Reporting_Parameters` command to the IUT, with the following parameter values: `Connection_Handle` set to the active connection handle, `High_Threshold = TSPX_Path_Loss_Upper_Boundary`, `High_Hysteresis = 0x03 (3dB)`. 

*Figure 4.485: [Path Loss Monitoring]*
Low_Threshold = TSPX_Path_Loss_Lower_Boundary, Low_Hysteresis = 0x03 (3dB), Min_Time_Spent = 0x01. The IUT responds with an HCI_Command_Complete.

3. The Upper Tester enables path loss reporting for the active connection by sending an HCI_LE_Set_Path_Loss_Reporti

4. The IUT sends an LL_POWER_CONTROL_REQ PDU, with Delta = 0 and PHY set to the active connection to the Lower Tester, querying the transmit power level.

5. The Lower Tester sends an LL_POWER_CONTROL_RSP PDU with the current local transmit power level.

6. The IUT sends anHCI_LE_Path_Loss_Threshold event, with Zone_Entered = 0x01 (Entered middle zone), to the Upper Tester.

7. The system under test is adjusted such that the IUT’s apparent path loss is gradually decreased until the IUT sends an HCI_LE_Path_Loss_Threshold event, with Zone_Entered = 0x00 (Entered low zone), to the Upper Tester.

8. The system under test is adjusted such that the IUT’s apparent path loss is gradually increased until the IUT sends an HCI_LE_Path_Loss_Threshold event, with Zone_Entered = 0x01 (Entered middle zone), after the path loss changes from the low zone to the middle zone.

9. The system under test is adjusted such that the IUT’s apparent path loss is gradually increased until the IUT sends an HCI_LE_Path_Loss_Threshold event, with Zone_Entered = 0x02 (Entered high zone) and the Current_Path_Loss, after the path loss changes from the middle zone to the high zone.

10. Move the high threshold above the current path loss by having the Upper Tester send an HCI_LE_Set_Path_Loss_Reporting_Parameters command to the IUT, with the following parameter values: Connection_Handle set to the active connection handle, High_Threshold = Current_Path_Loss from step 9 + 0x05, High_Hysteresis = 0x01 (1dB), Low_Threshold = TSPX_Path_Loss_Lower_Boundary, Low_Hysteresis = 0x01 (1dB), Min_Time_Spent = 0x01. The IUT responds with an HCI_Command_Complete.

11. The IUT sends an HCI_LE_Path_Loss_Threshold event with Zone_Entered = 0x01 (Entered middle zone) to the Upper Tester.

• Note

Note that Section 4.13.1 RF Test Conditions provides flexibility in how the IUT’s receive power is adjusted, and the means by which the apparent Path Loss is induced in the IUT may vary. An initial condition shall be chosen such that the apparent Path Loss as seen by the IUT can be varied across the supported Middle and Low Zone boundary.

• Expected Outcome

Pass Verdict

In steps 7, 8, and 9, the IUT sends an HCI_LE_Path_Loss_Threshold event to the Upper Tester, as described in each step.

In step 11, the IUT sends an HCI_LE_Path_Loss_Threshold event with Zone_Entered = 0x01 to the Upper Tester after the High Threshold zone was moved below the current Path Loss.

4.13.3.2 [Power Control Request – Initiate]

• Test Purpose

Verify that the IUT can request to adjust the Lower Tester’s transmit power level on a specified PHY on the default LE-ACL logical link. The tester will verify that the IUT properly handles each response from the request.
• Reference

[14] Section 5.1.17

• Initial Condition

Parameter: Golden_Range_Midpoint is the average value between the TSPX_Golden_Range_Upper and the TSPX_Golden_Range_Lower values (declared as IXIT).

- The Lower Tester and IUT are configured as specified in Section 4.13.1 RF Test Conditions.
- The Lower Tester begins transmitting with the power level 10 dB below TSPX_Golden_Range_Upper (declared as IXIT), or at the Golden_Range_Midpoint value, whichever is greater.
- An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration.

• Test Procedure
1. Repeat steps 2 and 3 until the IUT requests a power decrease by sending an LL_POWER_CONTROL_REQ PDU for each PHY in Table 4.156 with a negative Delta value or the resulting power increase would exceed 7 dB over TSPX_Golden_Range_Upper.
2. The Lower Tester increases its power by 2 dB at a rate of not more than 1 step per 5-second period.
3. The Lower Tester continuously transmits data packets over the ACL connection with a connection interval of 7.5 ms.
4. Repeat steps 5 and 6 until the IUT requests a power increase by sending an LL_POWER_CONTROL_REQ PDU for each PHY in Table 4.156 with a positive Delta value or the resulting power increase would exceed 7 dB under TSPX_Golden_Range_Lower.
5. The Lower Tester decreases its power by 2 dB at a rate of not more than 1 step per 5-second period.
6. The Lower Tester continuously transmits data packets over the ACL connection with a connection interval of 7.5 ms.

• Expected Outcome

Pass Verdict
The IUT sends an LL_POWER_CONTROL_REQ PDU to the Lower Tester requesting that it lowers its power.
Later, the IUT sends an LL_POWER_CONTROL_REQ PDU to the Lower Tester requesting that it increases its power.

Fail Verdict
The IUT fails to send an LL_POWER_CONTROL_REQ PDU to the Lower Tester, and the received power cannot increase without exceeding 7 dB over TSPX_Golden_Range_Upper.
The IUT fails to send an LL_POWER_CONTROL_REQ PDU to the Lower Tester, and the received power cannot decrease without falling 7 dB below TSPX_Golden_Range_Lower.

• Notes

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>PHY</th>
<th>IUT Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.2.1</td>
<td>LE 1M PHY (0x01)</td>
<td>Slave</td>
</tr>
<tr>
<td>4.13.3.2.2</td>
<td>LE 2M PHY (0x02)</td>
<td>Slave</td>
</tr>
<tr>
<td>4.13.3.2.3</td>
<td>LE Coded PHY S=8 (0x04)</td>
<td>Slave</td>
</tr>
<tr>
<td>4.13.3.2.3</td>
<td>LE Coded PHY S=2 (0x08)</td>
<td>Slave</td>
</tr>
<tr>
<td>4.13.3.2.4</td>
<td>LE 1M PHY (0x01)</td>
<td>Master</td>
</tr>
<tr>
<td>4.13.3.2.5</td>
<td>LE 2M PHY (0x02)</td>
<td>Master</td>
</tr>
</tbody>
</table>
### 4.13.3.3 [Power Control Response]

- **Test Purpose**
  Verify that the IUT can respond to a request from the Lower Tester to adjust the IUT’s transmit power level on a specified PHY on the default LE-ACL logical link. The Lower Tester will verify that the IUT properly responds to the request.

- **Reference**
  [14] Section 5.1.17

- **Initial Condition**
  An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration.

  The IUT’s list of supported Power Control Levels is defined by the TSPX_Supported_Power_Levels IXIT entry.
• Test Procedure

![Diagram of test procedure]

**Figure 4.487: [Power Control Response]**

1. If the TSPX_Supported_Power_Levels size has fewer than 3 supported Power Levels, the test ends with an Inconclusive Verdict.
2. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = -64 dB.
3. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester.
4. Repeat steps 2 and 3 until the IUT is at the lowest supported power level.
5. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command to the IUT with Connection_Handle and PHY set to the active connection.
6. The IUT sends an HCI_Command_Complete event to the Upper Tester with Status = 0x00 and Current_Transmit_Power_Level containing the actual transmit power level.
7. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = a value less than the difference between the next higher power level and the current power level.
8. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = the difference between the next higher power level and the Current_Transmit_Power_Level from step 6.
9. Repeat step 5.
10. The IUT sends an HCI_Command_Complete event to the Upper Tester, with Status = 0x00 and Current_Transmit_Power_Level containing the actual transmit power level, which must be equal
11. Repeat steps 7–10 until the IUT is at the highest supported power level.
12. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = a value less than the difference between the next lower power level and the current power level.
13. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 0.
14. Repeat step 5.
15. The IUT sends an HCI_Command_Complete event to the Upper Tester with Status = 0x00 and Current_Transmit_Power_Level containing the actual transmit power level, which must be equal to the value received in step 10.
16. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = a value between the difference between the next lower power level and current power level, and the difference between the power level lower than the next lower power level and current power levels.
17. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = the difference between the next lower power level and the Current_Transmit_Power_Level from step 15.
18. Repeat step 5.
19. The IUT sends an HCI_Command_Complete event to the Upper Tester with Status = 0x00 and Current_Transmit_Power_Level containing the actual transmit power level, which must be equal to the next lower power level from the Current_Transmit_Power_Level received in step 15.

• Expected Outcome

Pass Verdict
In steps 10, 15, and 19, the IUT reports the correct Current_Transmit_Power_Level.
In steps 8, 13, and 17, the IUT reports the correct Delta in the LL_POWER_CONTROL_RSP PDU.

Inconclusive Verdict
The IUT supports fewer than 3 Power Levels.

• Notes

While it may not be strictly necessary to attenuate the TX power from the IUT to the Lower Tester to execute this test, it may be advisable to ensure the RX power received by the Lower Tester is within the bounds of the Lower Tester’s RX operating range. A test setup with conditions similar to those described in Section 4.13.1 RF Test Conditions may be required.

The Lower Tester can receive LL_POWER_CHANGE_IND PDUs from the IUT for any currently managed PHYs other than the one requested.

Example:

Supported Power Levels: -10, 5, 10, 20. Below are the steps being executed.
1. IUT is at power level 17.
2. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU with Delta = -64 dB.
3. The IUT sends an LL_POWER_CONTROL_RSP PDU with Delta = -27.
4. Skip
5. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command.
6. The IUT sends an HCI_Command_Complete event with Current_Transmit_Power_Level = -10.
7. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU with Delta = 8 (less than 5 – (-10)).
8. The IUT sends an LL_POWER_CONTROL_RSP PDU with Delta = 15 (power level went from -10 to 5).
9. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command.
10. The IUT sends an HCI_Command_Complete event with Current_Transmit_Power_Level = 5.
11. Repeat the steps as below.
   - Repeat step 7 with Delta = 2 (less than 10 – 5).
   - Repeat step 8 with Delta = 5 (power level went from 5 to 10).
   - Repeat step 9.
   - Repeat step 10 with Current_Transmit_Power_Level = 10.
   - Repeat step 7 with Delta = 6 (less than 20 – 10).
   - Repeat step 8 with Delta = 10 (power level went from 10 to 20).
   - Repeat step 9.
   - Repeat step 10 with Current_Transmit_Power_Level = 20.
12. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU with Delta = -3 (a value between 0 and (10 – 20)).
13. The IUT sends an LL_POWER_CONTROL_RSP with Delta = 0 (power level didn’t change).
14. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command.
15. The IUT sends an HCI_Command_Complete event with Current_Transmit_Power_Level = 20.
16. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU with Delta = -12 (a value between (10 – 20) and (5 – 20)).
17. The IUT sends an LL_POWER_CONTROL_RSP PDU with Delta = -10 (power level went from 20 to 10).
18. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command.
19. The IUT sends an HCI_Command_Complete event with Current_Transmit_Power_Level = 10.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.3.1</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>4.13.3.3.2</td>
<td>LE 2M PHY (0x02)</td>
</tr>
<tr>
<td>4.13.3.3.3</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>4.13.3.3.4</td>
<td>LE Coded PHY S=2 (0x08)</td>
</tr>
<tr>
<td>4.13.3.3.5</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>4.13.3.3.6</td>
<td>LE 2M PHY (0x02)</td>
</tr>
</tbody>
</table>
### Test Case IDs

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.3.7</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>4.13.3.3.8</td>
<td>LE Coded PHY S=2 (0x08)</td>
</tr>
</tbody>
</table>

*Table 4.157: Test Cases*

#### 4.13.3.4 [Power Configuration Response, Min and Max Power Level Reached]

- **Test Purpose**
  
  Verify that the IUT properly handles the Lower Tester sending the LL_POWER_CHANGE_IND when the Lower Tester’s minimum or maximum power level has been reached.

- **Test Case IDs**
  
  LL/PCL/SLA/BV-12-C  
  LL/PCL/MAS/BV-12-C  

- **Reference**
  
  [14] Section 5.1.17

- **Initial Condition**
  
  Parameters: The Lower and Upper golden range are defined as TSPX_Golden_Range_Lower and TSPX_Golden_Range_Upper IIXIT values. Golden_Range_Midpoint is the average value between the TSPX_Golden_Range_Upper and the TSPX_Golden_Range_Lower values.
  
  - The Lower Tester and IUT are configured as specified in Section 4.13.1 RF Test Conditions.
  - The Lower Tester begins transmitting with the power level 10 dB above TSPX_Golden_Range_Lower, or at the Golden_Range_Midpoint value, whichever is less.
  - An ACL connection has been established between the IUT and the Lower Tester in the relevant role.
Test Procedure

An ACL Connection Has Been Established Between IUT and Lower Tester

1. The Upper Tester sends the HCI_LE_Set_Transmit_Power_Reporting_Enable command to the IUT with the Connection_Handle set to the ACL connection handle and Remote_Enable = 0x01 and receives a successful HCI_Command_Complete event.
2. The IUT sends an LL_POWER_CONTROL_REQ PDU to the Lower Tester with Delta = 0 and PHY = the current PHY and receives an LL_POWER_CONTROL_RSP PDU in return with the TxPower set to the current Remote transmit power.
3. The Lower Tester continuously transmits data packets over the ACL connection with a connection interval of 7.5 ms.
4. The Lower Tester increases its power by 2 dB, at a rate of not more than one step per 1-second period.

Figure 4.488: [Power Configuration Response, Min and Max Power Level Reached]
5. The Lower Tester sends an LL_POWER_CHANGE_IND PDU to the IUT with the PHY parameter set to the active connection, Min = 0, Delta and TxPower set to reflect the Power Change in step 4. If the Lower Tester is at the maximum power level, then Max = 1.

6. The IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with Reason = 0x01, Delta and Transmit_Power_Level set to reflect the values received in step 5. In step 5, if Max = 1, then the Transmit_Power_Level_Flag = 0b10.

7. Repeat steps 4–6 until the Lower Tester reaches maximum power.

8. Set the Lower Tester power level 10 dB below TSPX_Golden_Range_Upper, or at the Golden_Range_Midpoint value, whichever is greater.

9. The Lower Tester decreases its power by 2 dB, at a rate of not more than one step per 1-second period.

10. The Lower Tester sends an LL_POWER_CHANGE_IND PDU to the IUT with the PHY parameter set to the active connection, Max = 0, Delta and TxPower set to reflect the Power Change in step 9. If the Lower Tester is at the minimum power level, then Min = 1.

11. The IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with Reason = 0x01, Delta, PHY, and Transmit_Power_Level set to reflect the values received in step 10. In step 10, if Min = 1, then the Transmit_Power_Level_Flag = 0b01.

12. Repeat steps 9–11 until the Lower Tester reaches minimum power.

- Expected Outcome

  **Pass Verdict**

  In step 6, the IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with the Transmit_Power_Level received in step 5. The Transmit_Power_Level_Flag is set to 0b10 when Max = 1 in step 5.

  In step 11, the IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with the Transmit_Power_Level received in step 10. The Transmit_Power_Level_Flag is set to 0b01 when Min = 1 in step 10.

4.13.3.5  **[Set Acceptable Power Reduction Value]**

- **Test Purpose**

  Verify that the IUT sets an acceptable power reduction value in the LL_POWER_CONTROL_RSP PDU.

- **Test Case IDs**

  - LL/PCL/SLA/BV-16-C
  - LL/PCL/MAS/BV-16-C

- **Reference**

  [14] Section 5.1.17.1

- **Initial Condition**

  - An ACL connection has been established between the IUT and the Lower Tester in the relevant role.
  - The Lower Tester transmits to the IUT with sufficient power to ensure the IUT will have a non-zero APR.
• Test Procedure

![Diagram of LL Power Control Request and Response]

**Figure 4.489: [Set Acceptable Power Reduction Value]**

1. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU with Delta = 0.
2. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester, with APR set to a non-zero value.

• Expected Outcome

**Pass Verdict**

In step 2, the IUT sends an LL_POWER_CONTROL_RSP PDU with a non-zero APR value.

### 4.13.3.6 [Properly handle a Power Request PDU when waiting for a Power Response PDU]

• Test Purpose

Verify that when the IUT has sent out a Power Request PDU to the Lower Tester and receives a Power Request PDU from the Lower Tester while waiting for the Power Response PDU from the Lower Tester that the IUT properly sets the Acceptable Power Reduction parameter indicating it is unable to provide a value when responding to the Power Request PDU.

• Test Case IDs

**LL/PCL/SLA/BV-17-C**

**LL/PCL/MAS/BV-17-C**

• Reference

[14] Section 5.1.17

• Initial Condition


- The Lower Tester begins transmitting with the power level 10 dB above the declared TSPX_Golden_Range_Lower, or at the Golden_Range_Midpoint value, whichever is less.
- An ACL connection has been established between the IUT and the Lower Tester in the relevant role.
**Test Procedure**

1. The Lower Tester continuously transmits data packets over the ACL connection with a connection interval of 7.5 ms.
2. The Lower Tester decreases its power by 2 dB, at a rate of not more than one step per 5-second period.
3. Repeat steps 1 and 2 until the IUT sends a power increase request by sending an LL_POWER_CONTROL_REQ PDU with a positive Delta value, or until the Lower Tester reaches more than 6 dB below TSPX_Golden_Range_Lower.
4. The Lower Tester does not respond with LL_POWER_CONTROL_RSP PDU and shall send an LL_POWER_CONTROL_REQ PDU, with Delta set to a positive value.
5. The IUT sends an LL_POWER_CONTROL_RSP PDU with APR = 0xFF.

**Expected Outcome**

**Pass Verdict**

In step 3, the IUT sends an LL_POWER_CONTROL_REQ PDU with a positive Delta value.

In step 5, the IUT responds with an LL_POWER_CONTROL_RSP PDU with correctly set APR indicating the IUT is unable to provide an APR value.

**Fail Verdict**

The IUT fails to send an LL_POWER_CONTROL_REQ PDU to the Lower Tester and the received power cannot decrease without falling more than 6 dB below TSPX_Golden_Range_Lower.
4.13.3.7  [Power Control Request using an unsupported PHY]

• Test Purpose
  Verify that when the IUT receives a Power Control Request PDU from the Lower Tester for a PHY that the IUT does not support, the appropriate error is returned to the Lower Tester.

• Test Case IDs
  LL/PCL/SLA/BI-01-C
  LL/PCL/MAS/BI-01-C

• Reference
  [14] Section 5.1.17

• Initial Condition
  An ACL connection has been established between the IUT and the Lower Tester in the relevant role.

• Test Procedure

  1. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU, with PHY = 0x80 (Bit 7; currently RFU) and Delta = +6 dB.
  2. The IUT sends an LL_REJECT_EXT_IND PDU to the Lower Tester, with ErrorCode = 0x20.

• Expected Outcome

  Pass Verdict
  In step 2, the IUT sends LL_REJECT_EXT_IND PDU to the Lower Tester, with ErrorCode set to Unsupported LMP Parameter Value/Unsupported LL Parameter Value (0x20).

4.13.3.8  [Power Change Indication on PHY Change]

• Test Purpose
  Verify that the IUT sends a Power Change Indication on a PHY change.

• Reference
  [14] Section 5.1.18

• Initial Condition
  An ACL connection has been established on a 1M PHY between the IUT and the Lower Tester in the relevant role.
Test Procedure

1. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the TX_PHYS and RX_PHYS set to the current 1M PHY used in the ACL connection.

   **IUT is Slave**

2. The IUT sends an LL_PHY_RSP PDU to the Lower Tester.

3. The Lower Tester sends an LL_PHY_UPDATE_IND PDU to the IUT.

   **IUT is Master**

4. The IUT sends an LL_PHY_UPDATE_IND PDU to the Lower Tester.

5. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 0 and the PHY set to the PHY as specified in Table 4.158.

6. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with the TxPower as specified in Table 4.158.

7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the TX_PHYS and RX_PHYS set to the new PHY.
IUT is Slave

8. The IUT sends an LL_PHY_RSP PDU to the Lower Tester.
9. The Lower Tester sends an LL_PHY_UPDATE_IND PDU to the IUT.

IUT is Master

10. The IUT sends an LL_PHY_UPDATE_IND PDU to the Lower Tester.
11. After the PHY change, the IUT sends to the Upper Tester an HCI_LE_PHY_Update_Complete event, with Subevent_Code = 0x0C, Status = 0x00, TX_PHY and RX_PHY set to the new PHY.
12. If the TxPower parameter in the LL_POWER_CONTROL_RSP PDU in step 6 = 126 and the PHY specified in Table 4.158 = New PHY, then the IUT sends an LL_POWER_CHANGE_IND PDU to the Lower Tester with the PHY set to the new PHY and the TxPower set to the current Power. Optionally, the IUT will also send an LL_POWER_CHANGE_IND PDU to the Lower Tester with the PHY set to the 1M PHY.
13. If the TxPower parameter in the LL_POWER_CONTROL_RSP PDU in step 6 with the PHY specified in Table 4.158 = New Phy and the TxPower specified in Table 4.158 = the current power level, and the IUT changes its power level after the PHY change, then the IUT sends an LL_POWER_CHANGE_IND PDU to the Lower Tester with the PHY set to the new PHY and the TxPower set to the current Power.

• Expected Outcome

Pass Verdict

In step 12, the IUT sends the LL_POWER_CHANGE_IND PDU to the Lower Tester if the TxPower as specified in Table 4.158 is 126.

In step 12, the IUT does not send the LL_POWER_CHANGE_IND PDU to the Lower Tester if the TxPower as specified in Table 4.158 is the Current Power.

In step 13, the IUT sends the LL_POWER_CHANGE_IND PDU to the Lower Tester if the TxPower as specified in Table 4.158 is the Current Power and the IUT changes its power level after the PHY change.
Table 4.158: Power Change Indication on PHY Change Test Cases

4.13.3.9  [Power Control Request when a CIS has been established]

- **Test Purpose**
  Verify that the IUT can request to adjust the Lower Tester’s transmit power level when a Connected Isochronous Stream (CIS) is established. The tester will verify that the IUT properly handles each response from the request.

- **Reference**
  [14] Section 5.1.17
  Section 4.10 CIS

- **Initial Condition**
  Parameter: Golden_Range_Midpoint is the average value between the TSPX_Golden_Range_Upper and the TSPX_Golden_Range_Lower values (declared as IXIT).

  - The Lower Tester and IUT are configured as specified in Section 4.13.1 RF Test Conditions.
- The Lower Tester transmits with the power level 10 dB above TSPX_Golden_Range_Lower, or at the Golden_Range_Midpoint value, whichever is less.

- An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration. The ACL connection interval is 10 times the ISO interval for the corresponding PHY as specified in Section 4.10.2.1 [CIS Setup Procedure, Master Initiated] or, respectively, Section 4.10.3.1 [CIS Setup Response Procedure, Slave].

**Test Procedure**

1. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 0 and the PHY set to the ACL PHY.
2. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with the TxPower.
3. A **CIS Setup Procedure** is initiated (according to Section 4.10.2.1 [CIS Setup Procedure, Master Initiated] or, respectively, Section 4.10.3.1 [CIS Setup Response Procedure, Slave]), with the IUT in the role specified in the Test Case Configuration with PHY_M_To_S and PHY_S_To_M set to the CIS PHY specified in the Test Case Configuration. NOTE: The S=2 and S=8 Coded PHY CIS Setup Procedures utilize the LE Coded PHY test case configurations in Sections 4.10.2.1 and 4.10.3.1.
4. After the IUT sends the successful HCI_LE_CIS_Established event to the Upper Tester, the IUT sends an LL_POWER_CHANGE_IND PDU to the Lower Tester with the PHY set to the CIS PHY for each CIS PHY in Table 4.159.

5. The Lower Tester starts to continuously transmit PDUs data packets over the CIS connection.

6. The Lower Tester decreases its power by 2 dB, at a rate of not more than one step per 5-second period, on the CIS connection.

7. Repeat steps 5 and 6 until the IUT sends a power increase request by sending an LL_POWER_CONTROL_REQ PDU, or until the Lower tester reaches more than 6 dB below TSPX_Golden_Range_Lower.

8. The IUT sends an LL_POWER_CONTROL_REQ PDU with a positive Delta value and the PHY set to the CIS PHY.

9. The Lower Tester shall respond with a valid LL_POWER_CONTROL_RSP PDU, with Max = 1, Delta = 0.

10. Repeat steps 8 and 9 for each PHY in Table 4.159.

• Expected Outcome

  Pass Verdict
  In step 8, the IUT sends LL_POWER_CONTROL_REQ PDU to the Lower Tester, requesting power increase.

  Fail Verdict
  The IUT fails to send an LL_POWER_CONTROL_REQ PDU to the Lower Tester, and the received power cannot decrease without falling 6 dB below TSPX_Golden_Range_Lower.

• Notes

  The Lower Tester can receive LL_POWER_CHANGE_IND PDUs from the IUT for any currently managed PHYs other than the one requested.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>ACL PHY</th>
<th>CIS PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.9.1 LL/PCL/SLA/BV-20-C</td>
<td>Slave</td>
<td>LE 2M PHY (0x02)</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>[Power Control Request – LE 1M PHY – CIS, Initiate, Slave]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.13.3.9.2 LL/PCL/SLA/BV-22-C</td>
<td>Slave</td>
<td>LE 1M PHY (0x01)</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>[Power Control Request – LE Coded PHY – CIS, Initiate, Slave]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.13.3.9.3 LL/PCL/MAS/BV-20-C</td>
<td>Master</td>
<td>LE 2M PHY (0x02)</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>[Power Control Request – LE 1M PHY – CIS, Initiate, Master]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.13.3.9.4 LL/PCL/MAS/BV-23-C</td>
<td>Master</td>
<td>LE 1M PHY (0x01)</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>[Power Control Request – LE Coded PHY – CIS, Initiate, Master]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.159: Test Case
4.13.3.10  [Power Control Response when a CIS has been established]

- Test Purpose

Verify that the IUT can respond to a request from the Lower Tester to adjust the IUT’s transmit power level when a Connected Isochronous Stream (CIS) is established. The tester will verify that the IUT properly responds to the request.

- Reference

[14] Section 5.1.17

Section 4.10 CIS

- Initial Condition

An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration.

- Test Procedure

![Diagram of the test procedure]

1. A **CIS Setup Procedure** is initiated (according to Section 4.10.2.1 [CIS Setup Procedure, Master Initiated] or, respectively, Section 4.10.3.1 [CIS Setup Response Procedure, Slave]), with the IUT in the role specified in the Test Case Configuration with PHY_M_To_S and PHY_S_To_M set to
the PHY specified in the Test Case Configuration. NOTE: The S=2 and S=8 Coded PHY CIS Setup Procedures utilize the LE Coded PHY test case configurations in 4.10.2.1 and 4.10.3.1.

2. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 0 and the PHY set to the CIS PHY.
3. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester.
4. The Upper Tester sends an HCI_LE_Enhanced_Read_Transmit_Power_Level command to the IUT, with Connection_Handle and PHY set to the CIS PHY.
5. The IUT sends to the Upper Tester an HCI_Command_Complete event, with Status = 0x00 and Current_Transmit_Power_Level containing the transmit power level.
6. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU for the CIS connection, with Delta = –6 dB and the PHY set to the CIS PHY.
7. The IUT sends to the Lower Tester an LL_POWER_CONTROL_RSP PDU, with Delta set to an acceptable value around the requested change level.
8. Repeat step 4.
9. The IUT sends to the Upper Tester an HCI_Command_Complete event, with Status = 0x00 and Current_Transmit_Power_Level containing the transmit power level, which must be equal to the value received in step 4 plus the Delta value reported in step 6.
10. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU, with Delta = +6 dB and the PHY set to the CIS PHY.
11. The IUT sends to the Lower Tester an LL_POWER_CONTROL_RSP PDU, with Delta set to an acceptable value around the requested change level.
12. Repeat step 4.
13. The IUT sends to the Upper Tester an HCI_Command_Complete event, with Status = 0x00 and Current_Transmit_Power_Level containing the actual transmit power level, which must be equal to the value received in step 9 plus the Delta value reported in step 11.

• Expected Outcome

Pass Verdict

In steps 7 and 11, the IUT reports the correct Current_Transmit_Power_Level.

• Notes

While it may not be strictly necessary to attenuate the RF from the IUT to the Lower Tester to execute this test, it may be advisable to ensure the RX power received by the Lower Tester is within the bounds of the Lower Tester's RX operating range. Techniques similar to those described in Section 4.13.1 RF Test Conditions may be employed if required.

The Lower Tester can receive LL_POWER_CHANGE_IND PDUs from the IUT for any currently managed PHYS other than the one requested.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>ACL PHY</th>
<th>CIS PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.10.1 LL/PCL/SLA/BV-25-C [Power Control Response – LE 1M PHY – CIS, Slave]</td>
<td>Slave</td>
<td>LE 2M PHY (0x02)</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>4.13.3.10.2 LL/PCL/SLA/BV-28-C [Power Control Response – LE Coded PHY S=2 – CIS, Slave]</td>
<td>Slave</td>
<td>LE 1M PHY (0x01)</td>
<td>LE Coded PHY S=2 (0x08)</td>
</tr>
</tbody>
</table>
### Test Cases

**4.13.3.11 [Remote Transmit Power Level Request]**

- **Test Purpose**
  
  Verify that the IUT can request to read the transmit power level of the remote device. The tester will verify that the IUT properly handles each response from the request. Also verify that the IUT properly rejects the Lower Tester sending a power request with TxPower = 126.

- **Reference**
  
  [14] Section 5.1.17

- **Initial Condition**
  
  An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration.

- **Test Procedure**

  ![Diagram](Diagram.png)

  **Figure 4.495: [Remote Transmit Power Level Request]**

  1. The Upper Tester sends an HCI_LE_Read_Remote_Transmit_Power_Level command to the IUT, with Connection_Handle and PHY set to the active connection.
  2. The IUT sends to the Upper Tester an HCI_Command_Status event, with Status = 0x00.
  3. The IUT sends to the Lower Tester an LL_POWER_CONTROL_REQ PDU with Delta = 0. If the TxPower = 126, the test case ends with a fail verdict.
4. The Lower Tester sends an LL_POWER_CONTROL_RSP PDU with Delta = 0.
5. The IUT sends to the Upper Tester an HCI_LE_Transmit_Power_Reporting event with Status = 0x00, Reason = 0x02, Delta = 0 and the Transmit_Power_Level parameters set to the Lower Tester value.
6. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 0 and TxPower = 126.
7. The IUT sends an LL_REJECT_EXT_IND PDU to the Lower Tester with ErrorCode = 0x1E.

• **Expected Outcome**
  
  **Pass Verdict**
  In step 3, the IUT sends an LL_POWER_CONTROL_REQ PDU with Delta = 0.
  In step 5, the IUT reports the correct parameters to the Upper Tester.
  In step 7, the IUT sends an LL_REJECT_EXT_IND PDU to the Lower Tester with ErrorCode = 0x1E.

  **Fail Verdict**
  In step 3, the IUT sends an LL_POWER_CONTROL_REQ PDU with TxPower = 126.

• **Notes**

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.11.1 LL/PCL/SLA/BV-29-C [Remote Power Transmit Level Request – LE 1M PHY – Initiate, Slave]</td>
<td>Slave</td>
<td>LE 1M PHY (0x00)</td>
</tr>
</tbody>
</table>

*Table 4.161: Test Cases*

4.13.3.12  **[Power Control Response with RF Path Compensation]**

• **Test Purpose**
  
  Verify that the IUT can respond to a request from the Lower Tester to adjust the IUT’s transmit power level on a specified PHY on the default LE-ACL logical link, with RF path compensation. The Lower Tester will verify that the IUT properly responds to the request.

• **Reference**
  
  [14] Section 5.1.17

• **Initial Condition**
  
  - An ACL connection has been established between the IUT and the Lower Tester where the IUT role and the PHY used are specified in the Test Case Configuration.
• **Test Procedure**

Same as for Section 4.13.3.3 [Power Control Response], depending on the role and configuration in Table 4.163, with the following pre-condition:

- The Upper Tester sends an HCI_LE_Write_RF_Path_Compensation command to the IUT. The RF_Tx_Path_Compensation_Value parameter shall be set as shown in Table 4.162 for the current round.

Repeat the test procedure for each round shown in Table 4.162.

<table>
<thead>
<tr>
<th>Round</th>
<th>RF Tx Path Compensation Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 dB</td>
</tr>
<tr>
<td>2</td>
<td>+5 dB</td>
</tr>
<tr>
<td>3</td>
<td>–5 dB</td>
</tr>
<tr>
<td>4</td>
<td>0 dB</td>
</tr>
</tbody>
</table>

*Table 4.162: RF Tx Path Compensation Values*

• **Expected Outcome**

**Pass Verdict**


• Notes

While it may not be strictly necessary to attenuate the RF from the IUT to the Lower Tester to execute this test, it may be advisable to ensure the RX power received by the Lower Tester is within the bounds of the Lower Tester's RX operating range. Techniques similar to those described in Section 4.13.1 RF Test Conditions may be employed if required.

The Lower Tester can receive LL_POWER_CHANGE_IND PDUs from the IUT for any currently managed PHYs other than the one requested.

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Role</th>
<th>PHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13.3.12.1</td>
<td>Slave</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>4.13.3.12.2</td>
<td>Slave</td>
<td>LE 2M PHY (0x02)</td>
</tr>
<tr>
<td>4.13.3.12.3</td>
<td>Slave</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>4.13.3.12.4</td>
<td>Slave</td>
<td>LE Coded PHY S=2 (0x08)</td>
</tr>
<tr>
<td>4.13.3.12.5</td>
<td>Master</td>
<td>LE 1M PHY (0x01)</td>
</tr>
<tr>
<td>4.13.3.12.6</td>
<td>Master</td>
<td>LE 2M PHY (0x02)</td>
</tr>
<tr>
<td>4.13.3.12.7</td>
<td>Master</td>
<td>LE Coded PHY S=8 (0x04)</td>
</tr>
<tr>
<td>4.13.3.12.8</td>
<td>Master</td>
<td>LE Coded PHY S=2 (0x08)</td>
</tr>
</tbody>
</table>

Table 4.163: Test Cases

4.13.3.13 [Max and Min Power Level Response at Max and Min Power]

• Test Purpose

Verify that the IUT properly responds to an increase power request when the IUT is at maximum transmit power. Verify that the IUT properly responds to a decrease power request when the IUT is at minimum transmit power.
• Test Case IDs

LL/PCL/SLA/BV-40-C
LL/PCL/MAS/BV-40-C

• Reference

[14] Section 5.1.17

• Initial Condition

- Parameters: The Lower Tester and IUT are configured as specified in Section 4.13.1 RF Test Conditions.

- An ACL connection has been established between the IUT and the Lower Tester with the IUT Transmit Power between the maximum and the minimum transmit power.

• Test Procedure

![Diagram of test procedure]

Figure 4.496: [Max and Min Power Level Response at Max and Min Power]
1. The Upper Tester sends the HCI_LE_Set_Transmit_Power_Reporting_Enable command to the IUT with the Connection_Handle set to the ACL connection handle and Local_Enable = 0x01 and receives a successful HCI_Command_Complete event.
2. The Lower Tester continuously transmits data packets over the ACL connection with a connection interval of 7.5 ms.
3. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 2 and PHY = the current PHY.
4. The IUT increases its power by 2 dB.
5. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 2 and the TxPower set to the current Local transmit power. If the IUT is at maximum power, then Max = 1.
6. The IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with Reason = 0x00, Delta, Transmit_Power_Level set to the current local power level. If the IUT is at the maximum transmit power level, then Transmit_Power_Level_Flag = 0b10.
7. Repeat steps 3–6 every second until the IUT is at maximum power.
8. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = 2 and PHY = the current PHY.
9. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 0, Max = 1, and the TxPower set to the current Local transmit power.
10. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = −2 and PHY = the current PHY.
11. The IUT decreases its power by 2 dB.
12. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = −2 and the TxPower set to the current Local transmit power. If the IUT is at minimum power, then Min = 1.
13. The IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with Reason = 0x00, Delta, Transmit_Power_Level set to the current local power level. If the IUT is at the minimum transmit power level, then Transmit_Power_Level_Flag = 0b01.
14. Repeat steps 10–13 every second until the IUT is at maximum power.
15. The Lower Tester sends an LL_POWER_CONTROL_REQ PDU to the IUT with Delta = −2 and PHY = the current PHY.
16. The IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 0, Min = 1, and the TxPower set to the current Local transmit power.

• Expected Outcome

Pass Verdict

In step 9, the IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 0 and Max = 1.

In step 12, the IUT sends an LL_POWER_CONTROL_RSP PDU to the Lower Tester with Delta = 0 and Min = 1.

In steps 6 and 13, the IUT sends an HCI_LE_Transmit_Power_Reporting event to the Upper Tester with the proper Delta, Transmit_Power_Level, and Transmit_Power_Level_Flag parameters.
5 Test Case Mapping

The Test Case Mapping Table (TCMT) maps test cases to specific capabilities in the ICS. Profiles, protocols and services may define multiple roles, and it is possible that a product may implement more than one role. The product shall be tested in all roles for which support is declared in the ICS document. For products which support more than one role, a separate TCMT shall be filled out for each role, and separate tests shall be conducted for each role.

The columns for the TCMT are defined as follows:

**Item:** contains a y/x reference, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS Proforma for LL in [4]. If the item is defined with Protocol, Profile or Service abbreviation before y/x, the table and feature number referenced are defined in the abbreviated ICS proforma document.

**Feature:** recommended to be the primary feature defined in the ICS being tested or may be the test case name.

**Test Case(s):** the applicable test case identifiers required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported.

For purpose and structure of the ICS/IXIT proforma and instructions for completing the ICS/IXIT proforma refer to the Bluetooth ICS and IXIT proforma document.

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL 3/1</td>
<td>Non-Connectable Advertising Events</td>
<td>LL/DDI/ADV/BV-01-C</td>
</tr>
<tr>
<td>LL 3/1 AND LL 3/3</td>
<td>Advertising Data: Non-Connectable</td>
<td>LL/DDI/ADV/BV-03-C</td>
</tr>
<tr>
<td>LL 3/2</td>
<td>Undirected Advertising Events</td>
<td>LL/DDI/ADV/BV-02-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/3</td>
<td>Advertising Data: Undirected</td>
<td>LL/DDI/ADV/BV-04-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/6</td>
<td>Scan Request: Undirected Connectable</td>
<td>LL/DDI/ADV/BV-05-C</td>
</tr>
<tr>
<td></td>
<td>Scan Request Invalid Address</td>
<td>LL/ENC/ADV/B-01-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/6 AND LL 3/8</td>
<td>Scan Request Device Filtering</td>
<td>LL/DDI/ADV/BV-08-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/6 AND LL 3/8 AND LL 6/1</td>
<td>Advertising With Static Address</td>
<td>LL/SEC/ADV/BV-01-C</td>
</tr>
<tr>
<td>LL 3/1 AND LL 2/4</td>
<td>Privacy – Non-connectable Undirected Advertising with private address</td>
<td>LL/SEC/ADV/BV-02-C, LL/SEC/ADV/BV-03-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 3/5 AND LL 2/6</td>
<td>Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Scannable Advertising</td>
<td>LL/SEC/ADV/BV-15-C</td>
</tr>
<tr>
<td>LL 3/5 AND LL 2/6 AND LL 2/5</td>
<td>Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA</td>
<td>LL/SEC/ADV/BV-21-C</td>
</tr>
<tr>
<td>LL 3/5 AND LL 2/7</td>
<td>Device Privacy Mode, Scannable Advertising</td>
<td>LL/SEC/ADV/BV-18-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 2/6</td>
<td>Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Undirected Connectable Advertising</td>
<td>LL/SEC/ADV/BV-16-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 2/7</td>
<td>Device Privacy Mode, Undirected Connectable Advertising</td>
<td>LL/SEC/ADV/BV-19-C</td>
</tr>
<tr>
<td>LL 3/4 AND LL 2/6</td>
<td>Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Directed Connectable Advertising</td>
<td>LL/SEC/ADV/BV-17-C</td>
</tr>
<tr>
<td>LL 3/4 AND LL 2/7</td>
<td>Device Privacy Mode, Directed Connectable Advertising</td>
<td>LL/SEC/ADV/BV-20-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7</td>
<td>Connection Request Invalid CRC Connection Request</td>
<td>LL/DDI/ADV/BI-02-C, LL/DDI/ADV/BV-06-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7 AND LL 3/6</td>
<td>Scan Request Connection Request</td>
<td>LL/DDI/ADV/BV-07-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7 AND LL 6/1 AND LL 6/2 AND LL 8/3</td>
<td>Accepting Connections With Hop Lengths</td>
<td>LL/FRH/ADV/BV-01-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7 AND LL 3/4</td>
<td>Directed Advertising Events</td>
<td>LL/DDI/ADV/BV-11-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7 AND LL 3/8</td>
<td>Connection Request Device Filtering</td>
<td>LL/DDI/ADV/BV-09-C</td>
</tr>
<tr>
<td>LL 3/2 AND LL 3/7 AND LL 6/16</td>
<td>Accepting Connections Timeout</td>
<td>LL/CON/ADV/BV-02-C</td>
</tr>
<tr>
<td>LL 3/4 AND LL 3/7 AND LL 6/1 AND LL 6/2</td>
<td>Directed Advertising Connection</td>
<td>LL/CON/ADV/BV-04-C</td>
</tr>
<tr>
<td>LL 3/5</td>
<td>Scannable Advertising Events</td>
<td>LL/DDI/ADV/BV-15-C</td>
</tr>
<tr>
<td>LL 3/5 AND LL 3/3</td>
<td>Advertising Data: Scannable</td>
<td>LL/DDI/ADV/BV-16-C</td>
</tr>
<tr>
<td>LL 3/5 AND LL 3/6</td>
<td>Scan Request: Scannable</td>
<td>LL/DDI/ADV/BV-17-C</td>
</tr>
<tr>
<td>LL 3/4a AND LL 3/7</td>
<td>Low Duty Cycle Directed Advertising Events</td>
<td>LL/DDI/ADV/BV-19-C</td>
</tr>
<tr>
<td>LL 3/6 AND (LL 3/2 OR LL 3/5)</td>
<td>Scan Request Invalid CRC</td>
<td>LL/DDI/ADV/BI-01-C</td>
</tr>
<tr>
<td>LL 3/7 AND LL 8/3 AND NOT LL 9/10</td>
<td>Accepting Connection Requests, Support Data channel selection algorithm</td>
<td>LL/CON/ADV/BV-11-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 3/9 AND LL 9/7</td>
<td>Extended Advertising – LE 2M PHY</td>
<td>LL/DDI/ADV/BV-49-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/12</td>
<td>Extended Advertising, Sending Tx Power in Advertisements</td>
<td>LL/DDI/ADV/BV-34-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/5 AND SUM ICS 21/16</td>
<td>Extended Advertising, Scannable, without ADI</td>
<td>LL/DDI/ADV/BV-25-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/5 AND NOT SUM ICS 21/16</td>
<td>Extended Advertising, Scannable, with ADI</td>
<td>LL/DDI/ADV/BV-45-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/5 AND SUM ICS 21/16 AND LL 9/7</td>
<td>Extended Advertising, Scannable, without ADI, LE 2M PHY</td>
<td>LL/DDI/ADV/BV-51-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/5 AND NOT SUM ICS 21/16 AND LL 9/7</td>
<td>Extended Advertising, Scannable, with ADI, LE 2M PHY</td>
<td>LL/DDI/ADV/BV-52-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/5 AND NOT SUM ICS 21/16 AND LL 9/9</td>
<td>Extended Advertising, Scannable, with ADI, LE Coded PHY</td>
<td>LL/DDI/ADV/BV-54-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/3</td>
<td>Extended Advertising, Legacy PDUs with Data</td>
<td>LL/DDI/ADV/BV-21-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/7</td>
<td>Extended Advertising, Connectable</td>
<td>LL/CON/ADV/BV-05-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/7 AND LL 9/7</td>
<td>Extended Advertising, Connectable, LE 2M PHY</td>
<td>LL/CON/ADV/BV-12-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/3 AND LL 9/10</td>
<td>Extended Advertising, Legacy PDUs with Data, CSA #2</td>
<td>LL/DDI/ADV/BV-22-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/3 AND NOT LL 9/10</td>
<td>Extended Advertising, Legacy PDUs with Data, CSA #1</td>
<td>LL/DDI/ADV/BV-50-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/7</td>
<td>Extended Advertising, Legacy PDUs, Connectable</td>
<td>LL/CON/ADV/BV-06-C</td>
</tr>
<tr>
<td>LL 3/9 AND HCI 5/36 AND NOT SUM ICS 21/16</td>
<td>LE Set Extended Advertising Data Command</td>
<td>LL/DDI/ADV/BI-05-C</td>
</tr>
<tr>
<td>LL 3/9 AND HCI 5/38 AND NOT SUM ICS 21/16</td>
<td>LE Set Extended Scan Response Data Command</td>
<td>LL/DDI/ADV/BI-06-C</td>
</tr>
<tr>
<td>LL 2/2 AND LL 2/5 AND LL 3/9 AND LL 3/7</td>
<td>Extended Advertising, Accepting Connections with Random address, LE 1M PHY</td>
<td>LL CON/ADV/BV-14-C</td>
</tr>
<tr>
<td>LL 2/2 AND LL 2/5 AND LL 3/9 AND LL 3/7 AND LL 9/7</td>
<td>Extended Advertising, Accepting Connections with Random address, LE 2M PHY</td>
<td>LL CON/ADV/BV-15-C</td>
</tr>
<tr>
<td>LL 2/2 AND LL 2/5 AND LL 3/9 AND LL 3/7 AND LL 9/9</td>
<td>Extended Advertising, Accepting Connections with Random address, LE Coded PHY</td>
<td>LL CON/ADV/BV-16-C</td>
</tr>
<tr>
<td>LL 3/10</td>
<td>Periodic Advertising</td>
<td>LL/DDI/ADV/BV-26-C</td>
</tr>
<tr>
<td>LL 3/10 AND LL 9/7</td>
<td>Periodic Advertising, LE 2M PHY</td>
<td>LL/DDI/ADV/BV-55-C</td>
</tr>
<tr>
<td>LL 3/10 AND LL 9/9</td>
<td>Periodic Advertising, LE Coded PHY</td>
<td>LL/DDI/ADV/BV-56-C</td>
</tr>
<tr>
<td>LL 3/10 AND LL 3/11</td>
<td>Sending Channel Map Update Indication in ACAD</td>
<td>LL/DDI/ADV/BV-62-C</td>
</tr>
<tr>
<td>LL 3/10 AND LL 3/11</td>
<td>Periodic Advertising, Multiple Sets</td>
<td>LL/DDI/ADV/BV-33-C</td>
</tr>
<tr>
<td>LL 3/11 AND LL 9/22</td>
<td>Extended Advertising, Multiple Sets, LE 1M PHY, 1 µs slots</td>
<td>LL/DDI/ADV/BV-49-C</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/22</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, 1µs Antenna Switching During Constant Tone Extension Transmission (AoD)</td>
<td>LL/DDI/ADV/BV-58-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/18</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD)</td>
<td>LL/DDI/ADV/BV-36-C</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/18 AND LL 9/7</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD), LE 2M PHY</td>
<td>LL/DDI/ADV/BV-57-C</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/22 AND LL 9/7</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, 1µs Antenna Switching During Constant Tone Extension Transmission (AoD), LE 2M PHY</td>
<td>LL/DDI/ADV/BV-59-C</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/19</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, No Antenna Switching During Constant Tone Extension Transmission (AoA)</td>
<td>LL/DDI/ADV/BV-37-C</td>
</tr>
<tr>
<td>LL 2/1 AND LL 3/13 AND LL 9/19 AND LL 9/7</td>
<td>Connectionless CTE Transmitter, Public Device Addresses, No Antenna Switching During Constant Tone Extension Transmission (AoA), LE 2M PHY</td>
<td>LL/DDI/ADV/BV-60-C</td>
</tr>
<tr>
<td>LL 3/9 AND LL 3/12 AND NOT (SUM ICS 21/14 OR SUM ICS 21/16)</td>
<td>Extended Advertising, Sending TxPower in Advertisements</td>
<td>LL/DDI/ADV/BV-61-C</td>
</tr>
<tr>
<td>LL 3/10 AND NOT SUM ICS 21/16</td>
<td>Periodic Advertising, SyncInfo Validation</td>
<td>LL/DDI/ADV/BV-43-C</td>
</tr>
<tr>
<td>LL 11/2</td>
<td>Connectionless CTE Receiver, Public Addresses</td>
<td>LL/DDI/SCN/BV-29-C LL/DDI/SCN/BV-31-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/22</td>
<td>Connectionless CTE Receiver, Public Addresses, 1 µs slots</td>
<td>LL/DDI/SCN/BV-49-C LL/DDI/SCN/BV-55-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/7 AND LL 9/22</td>
<td>Connectionless CTE Receiver, Public Addresses, LE 2M PHY, 1 µs slots</td>
<td>LL/DDI/SCN/BV-50-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/7</td>
<td>Connectionless CTE Receiver, Public Addresses, LE 2M PHY</td>
<td>LL/DDI/SCN/BV-48-C, LL/DDI/SCN/BV-54-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/7 AND LL 9/22</td>
<td>Connectionless CTE Receiver, Public Addresses, LE 2M PHY, 1 µs slots</td>
<td>LL/DDI/SCN/BV-56-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/21</td>
<td>Connectionless CTE Receiver, Public Addresses, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)</td>
<td>LL/DDI/SCN/BV-30-C, LL/DDI/SCN/BV-32-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/22</td>
<td>Connectionless CTE Receiver, Public Addresses, µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), 1 µs slots</td>
<td>LL/DDI/SCN/BV-58-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/24</td>
<td>Connectionless CTE Receiver, Public Addresses, 1µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)</td>
<td>LL/DDI/SCN/BV-52-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/21 AND LL 9/7</td>
<td>Connectionless CTE Receiver, Public Addresses, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/DDI/SCN/BV-51-C, LL/DDI/SCN/BV-57-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/22 AND LL 9/7</td>
<td>Connectionless CTE Receiver, Public Addresses, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY, 1 µs slots</td>
<td>LL/DDI/SCN/BV-59-C</td>
</tr>
<tr>
<td>LL 11/2 AND LL 9/24 AND LL 9/7</td>
<td>Connectionless CTE Receiver, Public Addresses, 1µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/DDI/SCN/BV-53-C</td>
</tr>
<tr>
<td>LL 4/1</td>
<td>Passive Scanning Invalid Address</td>
<td>LL/ENC/SCN/BI-01-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 4/1 AND LL 4/2</td>
<td>Passive Scanning Invalid CRC</td>
<td>LL/DDI/SCN/BI-02-C</td>
</tr>
<tr>
<td></td>
<td>Passive Scanning: Non-connectable</td>
<td>LL/DDI/SCN/BV-01-C</td>
</tr>
<tr>
<td></td>
<td>Passive Scanning: Undirected Events</td>
<td>LL/DDI/SCN/BV-10-C</td>
</tr>
<tr>
<td></td>
<td>Passive Scanning: Directed Events</td>
<td>LL/DDI/SCN/BV-11-C</td>
</tr>
<tr>
<td></td>
<td>Passive Scanning: Discoverable Events</td>
<td>LL/DDI/SCN/BV-12-C</td>
</tr>
<tr>
<td>LL 4/1 AND LL 4/2 AND LL 4/5</td>
<td>Passive Scanning Device Filtering</td>
<td>LL/DDI/SCN/BV-02-C</td>
</tr>
<tr>
<td>LL 4/2 AND LL 4/3</td>
<td>Active Scanning Invalid CRC</td>
<td>LL/DDI/SCN/BI-01-C</td>
</tr>
<tr>
<td></td>
<td>Active Scanning</td>
<td>LL/DDI/SCN/BV-03-C</td>
</tr>
<tr>
<td>LL 2/2 AND LL 4/2 AND LL 4/3</td>
<td>Random Address Scanning</td>
<td>LL/SEC/SCN/BV-01-C</td>
</tr>
<tr>
<td>LL 4/2 AND LL 4/3 AND LL 4/4</td>
<td>Scanning For Advertiser Types</td>
<td>LL/DDI/SCN/BV-05-C</td>
</tr>
<tr>
<td>LL 4/3</td>
<td>Active Scanning Invalid Address</td>
<td>LL/ENC/SCN/BI-02-C</td>
</tr>
<tr>
<td>LL 4/3 AND LL 4/2 AND LL 4/5</td>
<td>Active Scanning Device Filtering</td>
<td>LL/DDI/SCN/BV-04-C</td>
</tr>
<tr>
<td>LL 4/7</td>
<td>Extended Scanning</td>
<td>LL/DDI/SCN/BV-19-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-23-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 9/7</td>
<td>Extended Scanning, LE 2M PHY</td>
<td>LL/DDI/SCN/BV-42-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 9/9</td>
<td>Extended Scanning, LE Coded PHY</td>
<td>LL/DDI/SCN/BV-43-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND SUM 21/16</td>
<td>Extended Scanning, Active, Core 5.0</td>
<td>LL/DDI/SCN/BV-20-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-24-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/SCN/BV-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-44-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-45-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND NOT SUM 21/16</td>
<td>Extended Scanning, Active, Core 5.1</td>
<td>LL/DDI/SCN/BV-64-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-65-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-62-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/SCN/BV-05-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND LL 9/7 AND SUM 21/16</td>
<td>Extended Scanning, Active, LE 2M PHY, Core 5.0</td>
<td>LL/TIM/SCN/BV-04-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND LL 9/7 AND NOT SUM 21/16</td>
<td>Extended Scanning, Active, LE 2M PHY, Core 5.1</td>
<td>LL/TIM/SCN/BV-06-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 4/7 AND LL 9/9 AND LL 4/3 AND SUM 21/16</td>
<td>Extended Scanning, Active, LE Coded PHY, Core 5.0</td>
<td>LL/TIM/SCN/BV-02-C, LL/TIM/SCN/BV-03-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 9/9 AND LL 4/3 AND NOT SUM 21/16</td>
<td>Extended Scanning, Active, LE Coded PHY, Core 5.1</td>
<td>LL/DDI/SCN/BV-66-C, LL/TIM/SCN/BV-07-C, LL/TIM/SCN/BV-08-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND LL 4/5 AND LL 2/4 AND SUM 21/16</td>
<td>Extended Scanning, Active, Filtering Policies, Generation of private addresses, Core 5.0</td>
<td>LL/DDI/SCN/BV-33-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 4/3 AND LL 4/5 AND LL 2/4 AND NOT SUM 21/16</td>
<td>Extended Scanning, Active, Filtering Policies, Generation of private addresses, Core 5.1</td>
<td>LL/DDI/SCN/BV-63-C</td>
</tr>
<tr>
<td>LL 4/8 OR LL 11/1 AND LL 9/7</td>
<td>Scanning for Periodic Advertising or Synchronizing to Periodic Advertising, LE 2M PHY</td>
<td>LL/DDI/SCN/BV-46-C</td>
</tr>
<tr>
<td>LL 4/8 AND LL 9/9</td>
<td>Scanning for Periodic Advertising or Synchronizing to Periodic Advertising, LE Coded PHY</td>
<td>LL/DDI/SCN/BV-47-C</td>
</tr>
<tr>
<td>LL 11/1 AND HCI 6/37</td>
<td>Synchronizing to Periodic Advertising LE Set Periodic Advertising Receive Enable Command</td>
<td>LL/DDI/SCN/BV-38-C</td>
</tr>
<tr>
<td>LL 11/1 AND HCI 6/37 AND LL 9/7</td>
<td>Synchronizing to Periodic Advertising LE Set Periodic Advertising Receive Enable Command, LE 2M PHY</td>
<td>LL/DDI/SCN/BV-60-C</td>
</tr>
<tr>
<td>LL 11/1 AND HCI 6/37 AND LL 9/9</td>
<td>Synchronizing to Periodic Advertising LE Set Periodic Advertising Receive Enable Command, LE Coded PHY</td>
<td>LL/DDI/SCN/BV-61-C</td>
</tr>
<tr>
<td>LL 6/26</td>
<td>Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Slave Role, LE 1M</td>
<td>LL/CON/SLA/BV-88-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>LL 6/26 AND LL 9/9</td>
<td>Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Slave Role, LE Coded</td>
<td>LL/CON/SLA/BV-89-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/SLA/BV-91-C</td>
</tr>
<tr>
<td>LL 6/28</td>
<td>Accepting Periodic Advertising Sync Transfer, Slave Role, LE 1M</td>
<td>LL/CON/SLA/BV-98-C</td>
</tr>
<tr>
<td>LL 6/28 AND LL 9/7 AND NOT LL 9/9</td>
<td>Accepting Periodic Advertising Sync Transfer, Slave Role, LE 2M</td>
<td>LL/CON/SLA/BV-100-C</td>
</tr>
<tr>
<td>LL 6/29</td>
<td>Sending Long Control PDUs, Slave Role, LE 1M</td>
<td>LL/CON/SLA/BV-112-C</td>
</tr>
<tr>
<td>LL 6/29 AND LL 9/7 AND NOT LL 9/9</td>
<td>Sending Long Control PDUs, Slave Role, LE 2M</td>
<td>LL/CON/SLA/BV-113-C</td>
</tr>
<tr>
<td>LL 6/29 AND LL 9/9</td>
<td>Sending Long Control PDUs, Slave Role, LE Coded</td>
<td>LL/CON/SLA/BV-114-C</td>
</tr>
<tr>
<td>LL 6/30</td>
<td>Receiving Long Control PDUs, Slave Role, LE 1M</td>
<td>LL/CON/SLA/BV-109-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 6/30 AND LL 9/7 AND NOT LL 9/9</td>
<td>Receiving Long Control PDUs, Slave Role, LE 2M</td>
<td>LL/CON/SLA/BV-110-C</td>
</tr>
<tr>
<td>LL 6/30 AND LL 9/9</td>
<td>Receiving Long Control PDUs, Slave Role, LE Coded</td>
<td>LL/CON/SLA/BV-111-C</td>
</tr>
<tr>
<td>LL 7/26</td>
<td>Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Master Role, LE 1M</td>
<td>LL/CON/MAS/BV-84-C</td>
</tr>
<tr>
<td>LL 7/29</td>
<td>Receiving Long Control PDUs, Master Role, LE 1M</td>
<td>LL/CON/MAS/BV-105-C</td>
</tr>
<tr>
<td>LL 7/29 AND LL 9/7 AND NOT LL 9/9</td>
<td>Receiving Long Control PDUs, Master Role, LE 2M</td>
<td>LL/CON/MAS/BV-106-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LL 7/29 AND LL 9/9</td>
<td>Receiving Long Control PDUs, Master Role, LE Coded</td>
<td>LL/CON/MAS/BV-107-C</td>
</tr>
<tr>
<td>LL 4/6a AND LL 4/8</td>
<td>Scanning for Periodic Advertising</td>
<td>LL/DDI/SCN/BV-34-C</td>
</tr>
<tr>
<td>LL 4/6a AND LL 4/8</td>
<td>Periodic Sync Establishment Filtering Policies</td>
<td></td>
</tr>
<tr>
<td>LL 5/1</td>
<td>Connection Initiation Invalid CRC</td>
<td>LL/CON/INI/BI-01-C, LL/CON/INI/BV-04-C</td>
</tr>
<tr>
<td>LL 5/1</td>
<td>Connection Initiation Timeout</td>
<td></td>
</tr>
<tr>
<td>TLL 5/1 AND LL 7/1 AND LL 7/2</td>
<td>Slave Packets Invalid CRC</td>
<td>LL/CON/INI/BI-02-C, LL/ENC/INI/BI-01-C</td>
</tr>
<tr>
<td>TLL 5/1 AND LL 7/1 AND LL 7/2</td>
<td>Slave Packets Invalid Address</td>
<td>LL/CON/INI/BV-03-C, LL/CON/INI/BV-01-C</td>
</tr>
<tr>
<td>TLL 5/1 AND LL 7/1 AND LL 7/2</td>
<td>Connection Initiation Missed Replies</td>
<td></td>
</tr>
<tr>
<td>TLL 5/1 AND LL 7/1 AND LL 7/2</td>
<td>Connection Initiation</td>
<td></td>
</tr>
<tr>
<td>LL 5/1 AND LL 5/3 AND LL 7/1 AND LL 7/2</td>
<td>Initiation Device Filtering: Undirected</td>
<td>LL/CON/INI/BV-06-C, LL/CON/INI/BV-07-C</td>
</tr>
<tr>
<td>LL 5/1 AND LL 5/3 AND LL 7/1 AND LL 7/2</td>
<td>Initiation Device Filtering: Directed</td>
<td></td>
</tr>
<tr>
<td>LL 5/1 AND LL 8/3 AND NOT LL 9/10</td>
<td>Requesting Connections, Support Data channel selection algorithm</td>
<td>LL/CON/INI/BV-22-C</td>
</tr>
<tr>
<td>LL 5/2 AND LL 7/1 AND LL 7/2</td>
<td>Connecting to Directed Advertising</td>
<td>LL/CON/INI/BV-02-C</td>
</tr>
<tr>
<td>LL 5/1 AND LL 2/4 AND LL 2/5</td>
<td>Privacy – Connection Establishment</td>
<td>LL/CON/INI/BV-08-C, LL/CON/INI/BV-09-C</td>
</tr>
<tr>
<td>LL 5/1 AND LL 2/4 AND LL 2/6</td>
<td>Generation of private addresses, Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Connection Establishment</td>
<td>LL/CON/INI/BV-18-C, LL/CON/INI/BV-24-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 5/1 AND LL 2/4 AND LL 2/7</td>
<td>Generation of private addresses, Device Privacy Mode, Connection Establishment</td>
<td>LL/CON/INI/BV-20-C</td>
</tr>
<tr>
<td>LL 5/2 AND LL 2/4 AND LL 2/5</td>
<td>Privacy – Connection Establishment with directed advertisement</td>
<td>LL/CON/INI/BV-10-C, LL/CON/INI/BV-11-C, LL/CON/INI/BV-12-C</td>
</tr>
<tr>
<td>LL 5/2 AND LL 2/4 AND LL 2/6</td>
<td>Generation of private addresses, Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Connection Establishment with directed advertisement</td>
<td>LL/CON/INI/BV-19-C</td>
</tr>
<tr>
<td>LL 5/2 AND LL 2/4 AND LL 2/7</td>
<td>Generation of private addresses, Device Privacy Mode, Connection Establishment with directed advertisement</td>
<td>LL/CON/INI/BV-21-C</td>
</tr>
<tr>
<td>LL 5/1 AND LL 2/6</td>
<td>Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled</td>
<td>LL/CON/INI/BV-23-C</td>
</tr>
<tr>
<td>LL 5/4</td>
<td>Connection initiation using extended advertising</td>
<td>LL/CON/INI/BV-13-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 5/1 AND LL 9/7</td>
<td>Extended Scanning, Connection Initiation, LE 2M PHY</td>
<td>LL/CON/INI/BV-25-C</td>
</tr>
<tr>
<td>LL 4/7 AND LL 5/1 AND LL 9/9</td>
<td>Extended Scanning, Connection Initiation, LE Coded PHY</td>
<td>LL/CON/INI/BV-26-C</td>
</tr>
<tr>
<td>LL 4/1 AND LL 4/6 AND LL 2/5</td>
<td>Privacy – Passive Scanning</td>
<td>LL/DDI/SCN/BV-13-C, LL/DDI/SCN/BV-14-C</td>
</tr>
<tr>
<td>LL 4/1 AND LL 2/6</td>
<td>Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Passive Scanning</td>
<td>LL/DDI/SCN/BV-26-C</td>
</tr>
<tr>
<td>LL 4/1 AND LL 2/7</td>
<td>Device Privacy Mode, Passive Scanning</td>
<td>LL/DDI/SCN/BV-28-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| LL 4/3 AND LL 2/4 AND LL 2/5 | Privacy – Active Scanning | LL/DDI/SCN/BV-15-C  
LL/DDI/SCN/BV-16-C  
LL/DDI/SCN/BV-17-C  
LL/DDI/SCN/BV-18-C  
LL/DDI/SCN/BI-03-C |
| LL 3/2 AND LL 3/6 | | LL/TIM/ADV/BV-01-C  
LL/TIM/ADV/BV-02-C |
| LL 6/1 | | LL/ENC/SLA/BI-01-C  
LL/TIM/SLA/BV-02-C  
LL/TIM/SLA/BV-03-C  
LL/TIM/SLA/BV-04-C |
| LL 6/1 AND LL 6/2 | | LL/CON/SLA/BI-01-C  
LL/CON/SLA/BV-02-C  
LL/CON/SLA/BV-15-C  
LL/CON/SLA/BV-16-C  
LL/CON/SLA/BV-17-C  
LL/CON/SLA/BV-18-C  
LL/TIM/SLA/BV-05-C |
| LL 6/1 AND LL 6/2 AND LL 6/19 | | LL/CON/SLA/BV-19-C  
LL/CON/SLA/BV-20-C |
| LL 6/1 AND LL 6/2 | | LL/CON/SLA/BV-21-C |
| LL 6/14 | | LL/CON/SLA/BI-02-C |
| LL 6/1 AND LL 6/2 AND LL 6/10 | | LL/CON/SLA/BV-10-C  
LL/TIM/SLA/BV-01-C |
| LL 6/1 AND LL 6/2 AND LL 6/11 | | LL/FRH/SLA/BV-01-C |
| LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/5 AND LL 6/6 | | LL/SEC/SLA/BV-01-C  
LL/SEC/SLA/BV-04-C |
<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/5 AND LL 6/6 AND LL 6/18</td>
<td></td>
<td>LL/SEC/SLA/BV-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/SEC/SLA/BV-03-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/7 AND LL 6/20</td>
<td></td>
<td>LL/TIM/SLA/BV-06-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/SLA/BV-07-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/13</td>
<td></td>
<td>LL/CON/SLA/BI-05-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/14</td>
<td></td>
<td>LL/CON/SLA/BV-11-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/15</td>
<td></td>
<td>LL/CON/SLA/BV-12-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/16</td>
<td></td>
<td>LL/CON/SLA/BV-13-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/3</td>
<td></td>
<td>LL/PAC/SLA/BV-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/PAC/SLA/BI-01-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/4</td>
<td></td>
<td>LL/CON/SLA/BV-14-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/5</td>
<td>Slave Transmitting Fragmented L2CAP Header</td>
<td>LL/CON/SLA/BV-04-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DFL/SLA/BV-01-C</td>
</tr>
<tr>
<td>LL 6/12</td>
<td></td>
<td>LL/SEC/SLA/BI-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/SEC/SLA/BI-04-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/SEC/SLA/BI-05-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/6 AND LL 6/5</td>
<td></td>
<td>LL/CON/SLA/BV-06-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/6</td>
<td>Slave Receiving Fragmented L2CAP Header</td>
<td>LL/CON/SLA/BV-05-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/DFL/SLA/BV-02-C</td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/11 AND LL 9/10</td>
<td>Slave Transmissions, Acknowledgement Scheme, Accepting Channel Map Update, Channel Selection Algorithm #2</td>
<td>LL/FRH/SLA/BV-02-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| LL 6/22 AND HCI 10/13 | Slave Data Length Update | LL/CON/SLA/BV-77-C  
LL/CON/SLA/BV-78-C  
LL/CON/SLA/BV-79-C  
LL/CON/SLA/BV-131-C  
LL/CON/SLA/BI-10-C |
| LL 6/22 AND LL 9/7 AND HCI 10/13 AND HCI 13/12 | Slave Data Length Update, LE 2M PHY | LL/CON/SLA/BV-80-C  
LL/CON/SLA/BV-81-C  
LL/CON/SLA/BV-129-C  
LL/CON/SLA/BI-11-C |
| LL 6/22 AND LL 9/9 AND HCI 10/13 AND HCI 13/12 | Slave Data Length Update, LE Coded PHY | LL/CON/SLA/BV-82-C  
LL/CON/SLA/BV-83-C  
LL/CON/SLA/BV-84-C  
LL/CON/SLA/BV-130-C  
LL/CON/SLA/BV-132-C  
LL/CON/SLA/BI-12-C |
| LL 7/22 AND HCI 10/13 | Master Data Length Update | LL/CON/MAS/BV-73-C  
LL/CON/MAS/BV-74-C  
LL/CON/MAS/BV-75-C  
LL/CON/MAS/BV-128-C  
LL/CON/MAS/BI-07-C |
| LL 7/22 AND LL 9/7 AND HCI 10/13 AND HCI 13/12 | Master Data Length Update, LE 2M PHY | LL/CON/MAS/BV-76-C  
LL/CON/MAS/BV-77-C  
LL/CON/MAS/BV-126-C  
LL/CON/MAS/BI-08-C |
| LL 7/22 AND LL 9/9 AND HCI 10/13 AND HCI 13/12 | Master Data Length Update, LE Coded PHY | LL/CON/MAS/BV-78-C  
LL/CON/MAS/BV-79-C  
LL/CON/MAS/BV-80-C  
LL/CON/MAS/BV-127-C  
LL/CON/MAS/BV-129-C  
LL/CON/MAS/BI-09-C |
| LL 6/24 AND LL 9/21 | Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) | LL/CON/SLA/BV-60-C  
LL/CON/SLA/BV-75-C |
<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL 1/4 AND LL 9/21 AND LL 9/7</td>
<td>Receive Unrequested Constant Tone Extension, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)</td>
<td>LL/CON/SLA/BV-69-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/21 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/CON/SLA/BV-122-C, LL/CON/SLA/BV-127-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/21 AND LL 9/7</td>
<td>Receive Unrequested Constant Tone Extension, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/CON/SLA/BV-124-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/21 AND LL 6/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start</td>
<td>LL/CON/SLA/BV-71-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/21 AND LL 6/12 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start, LE 2M PHY</td>
<td>LL/CON/SLA/BV-125-C</td>
</tr>
<tr>
<td>LL 6/25 AND LL 9/19</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role, No Antenna Switching During Constant Tone Extension Transmission (AoA)</td>
<td>LL/CON/SLA/BV-65-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LL 6/25 AND LL 9/19 AND LL 6/12</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role, No Antenna Switching During Constant Tone Extension Transmission (AoA) Encryption Start</td>
<td>LL/CON/SLA/BV-72-C</td>
</tr>
<tr>
<td>LL 6/25 AND (LL 9/18 OR LL 9/19)</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD), No Antenna Switching During Constant Tone Extension Transmission (AoA)</td>
<td>LL/CON/SLA/BV-66-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/20</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD)</td>
<td>LL/CON/SLA/BV-67-C LL/CON/SLA/BV-76-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/20</td>
<td>Receive Unrequested Constant Tone Extension, Slave Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD)</td>
<td>LL/CON/SLA/BV-70-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/20 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), LE 2M PHY</td>
<td>LL/CON/SLA/BV-121-C LL/CON/SLA/BV-123-C LL/CON/SLA/BV-128-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/20 AND LL 6/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start</td>
<td>LL/CON/SLA/BV-73-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/20 AND LL 6/12 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start, LE 2M PHY</td>
<td>LL/CON/SLA/BV-126-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LL 6/25 AND LL 9/18</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD)</td>
<td>LL/CON/SLA/BV-68-C</td>
</tr>
<tr>
<td>LL 6/25 AND LL 9/18 AND LL 6/12</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD), Encryption Start</td>
<td>LL/CON/SLA/BV-74-C</td>
</tr>
<tr>
<td>LL 6/25</td>
<td>Constant Tone Extension Request Procedure as Responder, Slave Role</td>
<td>LL/CON/SLA/BV-87-C</td>
</tr>
<tr>
<td>LL 6/24 AND LL 9/9 AND (LL 9/20 OR LL 9/21)</td>
<td>Constant Tone Extension Request Procedure as Initiator, Slave Role, LE Coded PHY, 2µs Antenna Sampling During Constant Tone Extension Reception</td>
<td>LL/CON/SLA/BV-134-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/9 AND (LL 9/20 OR LL 9/21)</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, LE Coded PHY, 2µs Antenna Sampling During Constant Tone Extension Reception</td>
<td>LL/CON/MAS/BV-131-C</td>
</tr>
<tr>
<td>LL 7/1</td>
<td></td>
<td>LL/TIM/MAS/BV-01-C, LL/CON/MAS/BV-02-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/10</td>
<td></td>
<td>LL/CON/MAS/BV-07-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/11</td>
<td>LL/FRH/MAS/BV-01-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6</td>
<td>LL/SEC/MAS/BV-01-C, LL/SEC/MAS/BV-03-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/12 AND LL 7/6 AND LL 7/19</td>
<td>LL/SEC/MAS/BV-05-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6 AND LL 7/19</td>
<td>LL/SEC/MAS/BV-02-C, LL/SEC/MAS/BI-07-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/12 AND LL 7/6</td>
<td>LL/SEC/MAS/BV-04-C, LL/SEC/MAS/BI-08-C, LL/SEC/MAS/BI-09-C</td>
<td></td>
</tr>
<tr>
<td>LL 6/1 AND LL 6/2 AND LL 6/5 AND LL 6/6 AND LL 6/12</td>
<td>LL/SEC/SLA/BV-05-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/13</td>
<td>LL/CON/MAS/BI-04-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/14</td>
<td>LL/CON/MAS/BV-08-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/15</td>
<td>LL/CON/MAS/BV-09-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/16</td>
<td>LL/CON/MAS/BV-10-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/3</td>
<td>LL/PAC/MAS/BV-01-C, LL/PAC/MAS/BI-01-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/4</td>
<td>LL/CON/MAS/BV-13-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/4 AND LL 7/13</td>
<td>LL/CON/MAS/BV-19-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/20</td>
<td>LL/CON/MAS/BV-20-C, LL/CON/MAS/BV-21-C</td>
<td></td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/5 AND LL 7/6</td>
<td>LL/CON/MAS/BV-22-C</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LL 7/14</td>
<td></td>
<td>LL/CON/MAS/BI-02-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/5 AND LL 7/6 AND LL 7/12 AND LL 7/16</td>
<td></td>
<td>LL/SEC/MAS/BI-01-C&lt;br&gt;LL/SEC/MAS/BI-03-C&lt;br&gt;LL/SEC/MAS/BI-04-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/5</td>
<td>Master Transmitting Fragmented L2CAP Header</td>
<td>LL/CON/MAS/BV-03-C&lt;br&gt;LL/DFL/MAS/BV-01-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/6 AND LL 7/12</td>
<td></td>
<td>LL/SEC/MAS/BI-05-C&lt;br&gt;LL/SEC/MAS/BI-06-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/6</td>
<td>Master Receiving Fragmented L2CAP Header</td>
<td>LL/CON/MAS/BV-04-C&lt;br&gt;LL/DFL/MAS/BV-02-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/6 AND LL 7/5</td>
<td></td>
<td>LL/CON/MAS/BV-05-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 9/10</td>
<td>Master Transmissions, Acknowledgement Scheme, Requesting Channel Map Update, Channel Selection Algorithm #2</td>
<td>LL/FRH/MAS/BV-02-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/23</td>
<td>Master Transmissions, Minimum Number Of Used Channels Procedure</td>
<td>LL/FRH/MAS/BV-03-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| LL 6/1 AND LL 6/2 AND LL 6/10a | Slave Initiating Connection Parameter Request | LL/CON/SLA/BV-24-C  
| | | LL/CON/SLA/BV-25-C  
| | | LL/CON/SLA/BV-26-C  
| | | LL/CON/SLA/BV-27-C  
| | | LL/CON/SLA/BV-85-C  
| | | LL/CON/SLA/BV-86-C  
| LL 6/1 AND LL 6/2 AND LL 6/10a AND LL 6/13 | Slave Initiating Connection Parameter Request – Timeout | LL/CON/SLA/BI-07-C  
| LL 6/1 AND LL 6/2 AND LL 6/10a AND LL 6/12 | Slave Initiating Connection Parameter Request – Collision With Encryption Start | LL/CON/SLA/BV-28-C  
| LL 6/1 AND LL 6/2 AND LL 6/10b | Slave Accepting Connection Parameter Request | LL/CON/SLA/BV-29-C  
| | | LL/CON/SLA/BV-30-C  
| | | LL/CON/SLA/BV-31-C  
| | | LL/CON/SLA/BV-32-C  
| | | LL/CON/SLA/BV-33-C  
| | | LL/CON/SLA/BI-08-C  
| | | LL/CON/SLA/BV-34-C  
| LL 7/1 AND LL 7/2 AND LL 7/4a | Master Responding to Feature Exchange | LL/CON/MAS/BV-23-C  
| LL 7/1 AND LL 7/2 AND LL 7/10a | Master Initiating Connection Parameter Request | LL/CON/MAS/BV-24-C  
| | | LL/CON/MAS/BV-25-C  
| | | LL/CON/MAS/BV-26-C  
| | | LL/CON/MAS/BV-29-C  
| | | LL/CON/MAS/BV-81-C  
| | | LL/CON/MAS/BV-82-C  
| LL 7/1 AND LL 7/2 AND LL 7/10a AND LL 7/13 | Master Initiating Connection Parameter Request – Timeout | LL/CON/MAS/BI-05-C  
| LL 7/1 AND LL 7/2 AND LL 7/10b | Master Accepting Connection Parameter Request | LL/CON/MAS/BV-30-C  
| | | LL/CON/MAS/BV-31-C  
| | | LL/CON/MAS/BV-32-C  
| | | LL/CON/MAS/BV-33-C  
| | | LL/CON/MAS/BV-34-C  
| | | LL/CON/MAS/BI-06-C  
| | | LL/CON/MAS/BV-27-C  
| | | LL/CON/MAS/BV-35-C  

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/10b AND LL 7/12</td>
<td>Master Accepting Connection Parameter Request – Collision with Encryption Start</td>
<td>LL/CON/MAS/BV-28-C</td>
</tr>
<tr>
<td>LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6 AND LL 9/4</td>
<td>Master Receiving LL_REJECT_EXT_IND</td>
<td>LL/SEC/MAS/BV-11-C</td>
</tr>
<tr>
<td>LL 7/12</td>
<td>Master Start Encryption: Overlapping Procedure; Unexpected Data Channel PDU</td>
<td>LL/SEC/MAS/BV-12-C, LL/SEC/MAS/BV-14-C</td>
</tr>
<tr>
<td>LL 7/12 AND LL 9/5</td>
<td>Master Start Encryption: Overlapping Procedure</td>
<td>LL/SEC/MAS/BV-13-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/21</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)</td>
<td>LL/CON/MAS/BV-56-C, LL/CON/MAS/BV-71-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/21</td>
<td>Receive Unrequested Constant Tone Extension, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)</td>
<td>LL/CON/MAS/BV-65-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/21 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/CON/MAS/BV-119-C, LL/CON/MAS/BV-124-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/21 AND LL 9/7</td>
<td>Receive Unrequested Constant Tone Extension, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA), LE 2M PHY</td>
<td>LL/CON/MAS/BV-121-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/21 AND LL 7/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start</td>
<td>LL/CON/MAS/BV-67-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/21 AND LL 7/12 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start, LE 2M PHY</td>
<td>LL/CON/MAS/BV-122-C</td>
</tr>
<tr>
<td>LL 7/25 AND LL 9/19</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, No Antenna Switching During Constant Tone Extension Transmission (AoA)</td>
<td>LL/CON/MAS/BV-61-C</td>
</tr>
<tr>
<td>LL 7/25 AND LL 9/19 AND LL 7/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, No Antenna Switching During Constant Tone Extension Transmission (AoA) Encryption Start</td>
<td>LL/CON/MAS/BV-68-C</td>
</tr>
<tr>
<td>LL 7/25 AND (LL 9/18 OR LL 9/19)</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD), No Antenna Switching During Constant Tone Extension Transmission (AoA)</td>
<td>LL/CON/MAS/BV-62-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/20</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD)</td>
<td>LL/CON/MAS/BV-63-C, LL/CON/MAS/BV-72-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/20</td>
<td>Receive Unrequested Constant Tone Extension, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD)</td>
<td>LL/CON/MAS/BV-66-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/20 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD). LE 2M PHY</td>
<td>LL/CON/MAS/BV-120-C, LL/CON/MAS/BV-125-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/20 AND LL 9/7</td>
<td>Receive Unrequested Constant Tone Extension, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), LE 2M PHY</td>
<td>LL/CON/MAS/BV-118-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/20 AND LL 7/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start</td>
<td>LL/CON/MAS/BV-69-C</td>
</tr>
<tr>
<td>LL 7/24 AND LL 9/20 AND LL 7/12 AND LL 9/7</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start, LE 2M PHY</td>
<td>LL/CON/MAS/BV-123-C</td>
</tr>
<tr>
<td>LL 7/25 AND LL 9/18</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD)</td>
<td>LL/CON/MAS/BV-64-C</td>
</tr>
<tr>
<td>LL 7/25 AND LL 9/18 AND LL 7/12</td>
<td>Constant Tone Extension Request Procedure as Initiator, Master Role, 2µs Antenna Switching During Constant Tone Extension Transmission (AoD), Encryption Start</td>
<td>LL/CON/MAS/BV-70-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>LL 7/25</td>
<td>Constant Tone Extension Request Procedure as Responder, Master Role</td>
<td>LL/CON/MAS/BV-83-C</td>
</tr>
<tr>
<td>LL 1/1 AND LL 9/7</td>
<td>LE 2M PHY</td>
<td>LL/DDI/ADV/BV-20-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/6a</td>
<td>PHY Update Procedure (multiple PHYs, Slave)</td>
<td>LL/CON/SLA/BV-44-C&lt;br&gt;LL/CON/SLA/BV-45-C&lt;br&gt;LL/CON/SLA/BV-48-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 6/23a</td>
<td>PHY Update Procedure (Slave)</td>
<td>LL/CON/SLA/BV-51-C&lt;br&gt;LL/CON/SLA/BI-09-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 6/11 AND LL 9/6a</td>
<td>Channel Map Collision (Slave Role)</td>
<td>LL/CON/SLA/BV-46-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 6/10b AND LL 9/6a</td>
<td>Parameter Update Collision (Slave Role)</td>
<td>LL/CON/SLA/BV-47-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/6 AND LL 9/7</td>
<td>PHY Update Procedure (LE 2M PHY, Master)</td>
<td>LL/CON/MAS/BV-49-C&lt;br&gt;LL/CON/MAS/BV-50-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/6a</td>
<td>PHY Update Procedure (Master)</td>
<td>LL/CON/MAS/BV-45-C&lt;br&gt;LL/CON/MAS/BV-46-C&lt;br&gt;LL/CON/MAS/BV-51-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 7/11 AND LL 7/23a</td>
<td>Channel Map Collision (Master Role)</td>
<td>LL/CON/MAS/BV-47-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 7/10b AND LL 7/23a</td>
<td>Parameter Update Collision (Master Role)</td>
<td>LL/CON/MAS/BV-48-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/6a AND NOT LL 9/8</td>
<td>PHY Update Procedure, Symmetric Links (Master)</td>
<td>LL/CON/MAS/BV-42-C&lt;br&gt;LL/CON/MAS/BV-44-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/6a AND LL 9/8</td>
<td>PHY Update Procedure, Symmetric and Asymmetric Links (Master)</td>
<td>LL/CON/MAS/BV-41-C&lt;br&gt;LL/CON/MAS/BV-43-C&lt;br&gt;LL/CON/MAS/BV-117-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 6/23a</td>
<td>PHY Update Procedure (Slave)</td>
<td>LL/CON/SLA/BV-40-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/6a AND NOT LL 9/8</td>
<td>Responding to PHY Update Procedure, Symmetric Links (Slave)</td>
<td>LL/CON/SLA/BV-43-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>LL1/4 AND LL 6/23a</td>
<td>Responding to PHY Update Procedure (Slave)</td>
<td>LL/CON/SLA/BV-42-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/9</td>
<td>PHY Update Procedure (LE Coded PHY, Slave)</td>
<td>LL/CON/SLA/BV-54-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/SLA/BV-57-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/6 AND LL 9/9</td>
<td>PHY Update Procedure (LE Coded PHY, Data Length Extension, Slave)</td>
<td>LL/CON/SLA/BV-55-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/SLA/BV-56-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/SLA/BV-58-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/SLA/BV-59-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/9</td>
<td>PHY Update Procedure (LE Coded PHY, Master)</td>
<td>LL/CON/MAS/BV-52-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/MAS/BV-55-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/6 AND LL 9/9</td>
<td>PHY Update Procedure (LE Coded PHY, Data Length Extension, Master)</td>
<td>LL/CON/MAS/BV-53-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CON/MAS/BV-54-C</td>
</tr>
<tr>
<td>LL 9/30 AND LL 6/31</td>
<td>Changeable Sleep Clock Accuracy, Slave</td>
<td>LL/TIM/SLA/BV-08-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/SLA/BV-10-C</td>
</tr>
<tr>
<td>LL 9/30</td>
<td>Sleep Clock Accuracy, Slave</td>
<td>LL/TIM/SLA/BV-09-C</td>
</tr>
<tr>
<td>LL 9/30 AND LL 7/31</td>
<td>Changeable Sleep Clock Accuracy, Master</td>
<td>LL/TIM/MAS/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/MAS/BV-05-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/MAS/BV-06-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/TIM/MAS/BV-07-C</td>
</tr>
<tr>
<td>LL 9/30</td>
<td>Sleep Clock Accuracy, Master</td>
<td>LL/TIM/MAS/BV-04-C</td>
</tr>
<tr>
<td></td>
<td><strong>Isochronous Streams Procedures</strong></td>
<td></td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>CIS Master Setup Procedure, Master Initiated – Master</td>
<td>LL/CIS/MAS/BV-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-26-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-30-C</td>
</tr>
<tr>
<td>(LL 7/32 OR LL 7/33) AND LL 9/7</td>
<td>CIS Master Setup Procedure, LE 2M PHY, Master Initiated – Master</td>
<td>LL/CIS/MAS/BV-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-27-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-31-C</td>
</tr>
<tr>
<td>(LL 7/32 OR LL 7/33) AND LL 9/9</td>
<td>CIS Master Setup Procedure, LE Coded PHY, Master Initiated – Master</td>
<td>LL/CIS/MAS/BV-25-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-28-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/CIS/MAS/BV-32-C</td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>Sending data in Unidirectional CIS – Master</td>
<td>LL/CIS/MAS/BV-05-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>LL 11/3</td>
<td>Receiving data in Unidirectional CIS – Master</td>
<td>LL/CIS/MAS/BV-06-C</td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>Sending and Receiving Data in Bidirectional – Master CIS</td>
<td>LL/CIS/MAS/BV-07-C</td>
</tr>
<tr>
<td>LL 7/33</td>
<td>Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Interleaved CIG – Master</td>
<td>LL/CIS/MAS/BV-08-C</td>
</tr>
<tr>
<td>LL 7/32</td>
<td>Sending and Receiving Data in Multiple CISes, Single CIG, Single Connection, Sequential – Master</td>
<td>LL/CIS/MAS/BV-09-C</td>
</tr>
<tr>
<td>LL 7/33</td>
<td>Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Interleaved CIG – Master</td>
<td>LL/CIS/MAS/BV-10-C</td>
</tr>
<tr>
<td>LL 7/33</td>
<td>Sending and Receiving Data in Multiple CISes, Single CIG, Multiple Connections, Sequential – Master</td>
<td>LL/CIS/MAS/BV-11-C</td>
</tr>
<tr>
<td>LL 9/31</td>
<td>Acknowledgement Scheme and Retransmissions – Master</td>
<td>LL/CIS/MAS/BV-13-C</td>
</tr>
<tr>
<td>LL 9/31 AND LL 7/12</td>
<td>Acknowledgement Scheme – Master – Encrypted</td>
<td>LL/CIS/MAS/BV-14-C</td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>Flushing of Packets in CIS – Master</td>
<td>LL/CIS/MAS/BV-17-C</td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>Bursting of Payloads in CIS – Master</td>
<td>LL/CIS/MAS/BV-18-C</td>
</tr>
<tr>
<td>LL 7/32 AND LL 7/33</td>
<td>Deterministic Packet Transmission in CIS – Master</td>
<td>LL/CIS/MAS/BV-19-C</td>
</tr>
<tr>
<td>LL 7/32 OR LL 7/33</td>
<td>CIS Terminate Procedure, Accepting – Slave</td>
<td>LL/CIS/MAS/BV-16-C</td>
</tr>
<tr>
<td>(LL 7/32 OR LL 7/33) AND LL 7/12</td>
<td>Set Encryption After CIS Established – Master</td>
<td>LL/CIS/MAS/BV-20-C</td>
</tr>
<tr>
<td>(LL 7/32 AND LL 7/33) AND LL 7/11</td>
<td>New Channel Map – Master</td>
<td>LL/CIS/MAS/BV-04-C</td>
</tr>
<tr>
<td>LL 6/32</td>
<td>CIS Setup Response Procedure, Slave</td>
<td>LL/CIS/SLA/BV-01-C</td>
</tr>
<tr>
<td>LL 9/32 AND LL 9/7</td>
<td>CIS Setup Response Procedure, Slave, LE 2M PHY</td>
<td>LL/CIS/SLA/BV-19-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LL 9/32</td>
<td>CIS Setup Response Procedure, Slave, Reject Response</td>
<td>LL/CIS/SLA/BV-02-C</td>
</tr>
<tr>
<td>LL 6/32 AND LL 9/32</td>
<td>Sending data in Unidirectional CIS – Slave</td>
<td>LL/CIS/SLA/BV-04-C</td>
</tr>
<tr>
<td>LL 9/32</td>
<td>Receiving data in Unidirectional CIS – Slave</td>
<td>LL/CIS/SLA/BV-05-C</td>
</tr>
<tr>
<td>LL 6/32 AND LL 9/32</td>
<td>Sending and Receiving Data in Bidirectional – Slave CIS</td>
<td>LL/CIS/SLA/BV-06-C</td>
</tr>
<tr>
<td>LL 9/32</td>
<td>Acknowledgement Scheme and Retransmissions – Slave</td>
<td>LL/CIS/SLA/BV-10-C, LL/CIS/SLA/BV-11-C</td>
</tr>
<tr>
<td>LL 9/32 AND LL 6/12</td>
<td>Acknowledgement Scheme – Slave – Encrypted</td>
<td>LL/CIS/SLA/BV-21-C</td>
</tr>
<tr>
<td>LL 6/32</td>
<td>Flushing of Packets in CIS – Slave</td>
<td>LL/CIS/SLA/BV-14-C</td>
</tr>
<tr>
<td>LL 6/32</td>
<td>Deterministic Packet Transmission in CIS – Slave</td>
<td>LL/CIS/SLA/BV-16-C</td>
</tr>
<tr>
<td>LL 9/32</td>
<td>CIS Terminate Procedure, Initiated – Slave</td>
<td>LL/CIS/SLA/BV-12-C</td>
</tr>
<tr>
<td>LL 9/32</td>
<td>CIS Terminate Procedure, Accepting – Slave</td>
<td>LL/CIS/SLA/BV-13-C</td>
</tr>
<tr>
<td>LL 6/11 AND LL 9/32</td>
<td>CIS Map Update – Slave</td>
<td>LL/CIS/SLA/BV-03-C</td>
</tr>
<tr>
<td>LL 12/2</td>
<td>Broadcasting a Broadcast Isochronous Stream, Unencrypted</td>
<td>LL/BIS/BRD/BV-01-C</td>
</tr>
<tr>
<td>LL 12/1 OR LL 12/2</td>
<td>Broadcasting a Broadcast Isochronous Stream</td>
<td>LL/BIS/BRD/BV-13-C, LL/BIS/BRD/BV-16-C</td>
</tr>
<tr>
<td>(LL 12/1 OR LL 12/2) AND LL 9/7</td>
<td>Broadcasting a Broadcast Isochronous Stream, LE 2M PHY</td>
<td>LL/BIS/BRD/BV-14-C</td>
</tr>
<tr>
<td>(LL 12/1 OR LL 12/2) AND LL 9/9</td>
<td>Broadcasting a Broadcast Isochronous Stream, LE Coded PHY</td>
<td>LL/BIS/BRD/BV-15-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>(LL 12/1 OR LL 12/2) AND GAP 10/5</td>
<td>BIG Channel Map Update, Broadcaster</td>
<td>LL/BIS/BRD/BV-09-C</td>
</tr>
<tr>
<td>LL 12/1 AND GAP 11b/1</td>
<td>Encrypting a Broadcast Isochronous Stream</td>
<td>LL/BIS/BRD/BV-02-C</td>
</tr>
<tr>
<td>LL 11/3</td>
<td>Synchronizing to a Broadcast Isochronous Stream</td>
<td>LL/BIS/SNC/BV-01-C</td>
</tr>
<tr>
<td>LL 11/4</td>
<td>Receiving Data on Broadcast Isochronous Streams</td>
<td>LL/BIS/SNC/BV-07-C, LL/BIS/SNC/BV-08-C, LL/BIS/SNC/BV-11-C, LL/BIS/SNC/BV-12-C</td>
</tr>
<tr>
<td>LL 9/34 AND GAP 16/4</td>
<td>BIG Channel Map Update, Synchronizer</td>
<td>LL/BIS/SNC/BV-09-C</td>
</tr>
<tr>
<td>LL 9/34</td>
<td>Receiving Control Information in a Broadcast Isochronous Stream</td>
<td>LL/BIS/SNC/BV-10-C</td>
</tr>
<tr>
<td>LL 11/4</td>
<td>Receiving Data in Multiple Broadcast Isochronous Streams</td>
<td>LL/BIS/SNC/BV-04-C</td>
</tr>
<tr>
<td>LL 11/4 AND GAP 4/1</td>
<td>Synchronizing to an encrypted Broadcast Isochronous Stream</td>
<td>LL/BIS/SNC/BV-02-C</td>
</tr>
<tr>
<td>LL 9/36 AND LL 9/31 AND LL 6/33</td>
<td>ISO Transmit Test Mode, CIS, Master</td>
<td>LL/IST/MAS/BV-01-C</td>
</tr>
<tr>
<td>LL 9/36 AND LL 9/32 AND LL 6/33</td>
<td>ISO Transmit Test Mode, CIS, Slave</td>
<td>LL/IST/SLA/BV-01-C</td>
</tr>
<tr>
<td>LL 1/6 AND LL 9/36</td>
<td>ISO Transmit Test Mode, BIS</td>
<td>LL/IST/BRD/BV-01-C</td>
</tr>
<tr>
<td>LL 1/6 AND LL 9/35</td>
<td>ISO Receive Test Mode, BIS</td>
<td>LL/IST/SNC/BV-01-C</td>
</tr>
<tr>
<td>LL 9/35 AND LL 9/31</td>
<td>ISO Receive Test Mode, CIS, Master</td>
<td>LL/IST/MAS/BV-03-C</td>
</tr>
<tr>
<td>LL 9/31 AND LL 9/30</td>
<td>CIS Updating SCA, Master</td>
<td>LL/CIS/MAS/BV-24-C</td>
</tr>
<tr>
<td>LL 9/31 AND NOT LL 9/30</td>
<td>Error Updating SCA, Master</td>
<td>LL/CIS/MAS/BI-01-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 9/32 AND NOT LL 9/30</td>
<td>Error Updating SCA, Slave</td>
<td>LL/CIS/SLA/BI-01-C</td>
</tr>
</tbody>
</table>

### Power Control Procedures

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL 1/4 AND LL 9/37</td>
<td>LE Power Control Request, Slave</td>
<td>LL/PCL/SLA/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/PCL/SLA/BV-08-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL/PCL/SLA/BV-33-C</td>
</tr>
<tr>
<td></td>
<td>LE Power Configuration Response, Max and Min, Slave</td>
<td>LL/PCL/SLA/BV-12-C</td>
</tr>
<tr>
<td></td>
<td>LE Power Control Remote Request, Slave</td>
<td>LL/PCL/SLA/BV-29-C</td>
</tr>
<tr>
<td></td>
<td>Acceptable Power Reduction, Slave</td>
<td>LL/PCL/SLA/BV-16-C</td>
</tr>
<tr>
<td></td>
<td>LL/PCL/SLA/BV-17-C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power Control Request, unsupported PHY, Slave</td>
<td>LL/PCL/SLA/BI-01-C</td>
</tr>
<tr>
<td></td>
<td>Max and Min Power Level Response at Max and Min Power, Master</td>
<td>LL/PCL/SLA/BV-40-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/37</td>
<td>LE Power Control Request, Master</td>
<td>LL/PCL/MAS/BV-03-C</td>
</tr>
<tr>
<td></td>
<td>LL/PCL/MAS/BV-08-C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LL/PCL/MAS/BV-33-C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LE Power Configuration Response, Max and Min, Master</td>
<td>LL/PCL/MAS/BV-12-C</td>
</tr>
<tr>
<td></td>
<td>Acceptable Power Reduction, Master</td>
<td>LL/PCL/MAS/BV-16-C</td>
</tr>
<tr>
<td></td>
<td>LL/PCL/MAS/BV-17-C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power Control Request, unsupported PHY, Master</td>
<td>LL/PCL/MAS/BI-01-C</td>
</tr>
<tr>
<td></td>
<td>Max and Min Power Level Response at Max and Min Power, Master</td>
<td>LL/PCL/MAS/BV-40-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/7 AND NOT LL 9/9 AND LL 9/37</td>
<td>LE Power Control Request, LE 2M PHY, Slave</td>
<td>LL/PCL/SLA/BV-34-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/7 AND LL 9/37</td>
<td>LE Power Control Request, LE 2M PHY, Slave</td>
<td>LL/PCL/SLA/BV-04-C, LL/PCL/SLA/BV-09-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/9 AND LL 9/37</td>
<td>LE Power Control Request, LE Coded PHY S=8, Slave</td>
<td>LL/PCL/SLA/BV-35-C</td>
</tr>
<tr>
<td></td>
<td>LE Power Control Request, LE Coded PHY S=2, Slave</td>
<td>LL/PCL/SLA/BV-36-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/7 AND NOT LL 9/9 AND LL 9/37</td>
<td>LE Power Control Request, LE 2M PHY, Master</td>
<td>LL/PCL/MAS/BV-34-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/7 AND LL 9/37</td>
<td>LE Power Control Request, LE 2M PHY, Master</td>
<td>LL/PCL/MAS/BV-04-C, LL/PCL/MAS/BV-09-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/39</td>
<td>LE Path Loss Monitoring, Slave</td>
<td>LL/PCL/SLA/BV-01-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/39</td>
<td>LE Path Loss Monitoring, Master</td>
<td>LL/PCL/MAS/BV-01-C</td>
</tr>
<tr>
<td>LL 1/5 AND LL 9/37 AND LL 9/7 AND LL 9/31</td>
<td>LE Power Control Request, CIS, Master</td>
<td>LL/PCL/MAS/BV-20-C, LL/PCL/MAS/BV-25-C</td>
</tr>
<tr>
<td>LL 1/4 AND LL 9/9 AND LL 9/37 AND LL 9/33</td>
<td>LE Power Control Request, LE Coded PHY S=8, Slave</td>
<td>LL/PCL/SLA/BV-22-C</td>
</tr>
<tr>
<td></td>
<td>LE Power Control Request, LE Coded PHY S=2, Slave</td>
<td>LL/PCL/SLA/BV-28-C</td>
</tr>
<tr>
<td></td>
<td>LE Power Control Request, LE Coded PHY S=2, Master</td>
<td>LL/PCL/MAS/BV-23-C</td>
</tr>
</tbody>
</table>

*Table 5.1: Test Case Mapping*
## 6 Revision History and Contributors

### Revision History

<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>D09r04</td>
<td>2009-05-06</td>
<td></td>
<td>Revision 09 approved by BTI (include review comments resolution).</td>
</tr>
<tr>
<td>D09r05</td>
<td>2009-06-09</td>
<td></td>
<td>Editorial review and markup.</td>
</tr>
<tr>
<td>D09r06</td>
<td>2009-10-30</td>
<td>*MSCs added.</td>
<td>*Test Procedures clarified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*Updated to LE Draft v1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*TP/SEC/SLA/BV-04-C and TP/SEC/MAS/BV-03-C added.</td>
</tr>
<tr>
<td>D09r08</td>
<td>2009-11-18</td>
<td></td>
<td>All changes accepted as agreed in LE Test group F2F the 18th</td>
</tr>
<tr>
<td>D09r07</td>
<td>2009-11-18</td>
<td>* Added comments from Magnus Sommansson.</td>
<td>* Added comments from Paul Vanoostende.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Typo errors updated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Errata 3127, 3221 incorporated.</td>
</tr>
<tr>
<td>LL.TS/Tokyo.1.0.0r1-15</td>
<td>2009-11-24-2009-12-15</td>
<td>Updated references</td>
<td>Editorial adjustments to harmonize with other Tokyo TS documents</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Incorporate errata 3299; affects 2 test cases: TP/SEC/SLA/BV-04C and TP/SEC/MAS/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Incorporate 3316 affects TP/CON/MAS/BV-08,09-C, TC/CON/SLA/BV-11,12-C: Modifications in MSCs and procedures, AND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/SLA/BI-05-C and TP/CON/MAS/BI-04-C, modified due to the inclusion of the T_Terminate timer in the termination procedure Modification of test cases, using updated control procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added TP/CON/SLA/BI-02-C and TP/CON/MAS/BI-02-C, to test the behavior of the IUT when it uses the T_Terminate timer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/SLA/BV-17-C:MSC corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/MAS/BV-16-C:MSC corrected</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/SLA/BI-04-C: Test case mapping corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/MAS/BI-02-C: MSC name corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/SLA/BV-10-C: Typo error in MSC corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TIM/SLA group, SEC/SLA/BI group, TP/SEC/MAS/BI-05, 06: MSCs and procedures corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Preamble MSC corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/INI/BI-02-C: ‘reason’ parameter value corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SEC group test cases updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SEC/SLA/BI-03-C and TP/SEC/SLA/BI-04-C: Verdict section corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test Case Mapping corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SEC/MAS/BI-03-C: MSC corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test procedure of TIM group test cases corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Change to final document number structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TPG integration issues with TCMTs corrected for TP/DDI/SCN/BV-12-C; TP/CON/SLA/BV-21-C; TP/CON/MAS/BV-16-C &amp; BV-17-C &amp; BV-18-C &amp; BV-22-C; TP/ENC/SCN/BI-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Corrected 8/2 to 7/2 in TP/SEC/MAS/BI-05-C; TP/SEC/MAS/BI-06-C</td>
</tr>
<tr>
<td>0</td>
<td>LL.TS/4.0.0</td>
<td>2009-12-17</td>
<td>Prepare for publication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOTE: In descending order:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3563: TP/DDI/ADV/BV-11-C: Updated initial condition and test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3559: TP/DDI/SCN/BV-05-C: Removed data fields from ADV_DIRECT_IND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3557: TP/SEC/SLA/BV-02-C: Update MSC and test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3549: TP/SEC/SCN/BV-01-C: Update MSC and test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3542: TP/DDI/ADV/BV-09-C: Update test procedure steps 7&amp;9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3532: TP/TIM/SLA/BV-04-C: Remove Note.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3524: TP/CON/SLA/BI-05-C, TP/CON/MAS/BI-04-C: add alternative MSCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3519: TP/SEC/MAS/BI-01-C: Correct MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3494: TP/SEC/SLA/BI-01-C: Update MSC and test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3493: TP/DDI/ADV/BV-09-C: Correct test procedure step 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3492: TP/CON/INI/BV-06-C, TP/CON/INI/BV-07-C: Correct test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3486: TP/DDI/SCN/BV-05-C: Add step to test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3472: TP/CON/INI/BI-02-C: Update purpose, MSC, test proc, pass verdict</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3471: TP/CON/INI/BI-02-C: update MSC, test proc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3470: TP/ENC/INI/BI-01-C: Update purpose, MSC, pass verdict</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3467: TP/CON/SLA/BV-04-C, TP/CON/MAS/BV-03-C: Update MSC and test procedures.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3460: TP/CON/SLA/BV-05-C: Modify MSC?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3410: TP/CON/INI/BV-01-C: Modify Reference, test procedure, and Pass verdict</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOTE: In ascending order: (in case of superseding TSEs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3371: TP/CON/SLA/BV-04-C: Update TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3374: TP/CON/SLA/BV-16-C, TP/CON/SLA/BV-17-C: Update MSCs and test procedures</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3375: TP/CON/MAS/BV-02-C: Update MSC latency value</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3376: TP/SEC/SLA/BV-02-C, TP/SEC/SLA/BV-03-C: Update MSCs and test procedures</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3377: TP/SEC/SLA/BV-04-C: Update Verdicts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3378: TP/DDI/ADV/BV-11-C: Correct MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3379: TP/CON/ADV/BI-01-C: Test purpose, caption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3380: TP/CON/SLA/BV-02-C: Update verdicts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3388: TP/FRH/ADV/BV-01-C: Correct TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3392: TP/PAC/MAS/BV-01-C: Correct TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3393: TP/CON/MAS/BV-03-C: Correct TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3394: TP/CON/SLA/BV-05-C: Correct TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3395: TP/CON/SLA/BV-06-C: Correct TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3399: TP/SEC/ADV/BV-01-C: Correct test purpose</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3400: TP/DDI/ADV/BV-15-C: Correct MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3401: TP/DDI/SCN/BV-01-C, TP/DDI/SCN/BV-10-C, TP/DDI/SCN/BV-11-C, TP/DDI/SCN/BV-12-C: Modify test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3403: TP/SEC/MAS/BV-02-C: Correct Fail verdict</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3404: TP/SEC/MAS/BI-06-C: Correct step 4, test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3405: TP/TIM/MAS/BV-01-C, TP/TIM/MAS/BV-02-C: Correct jitter values in test procedures</td>
</tr>
<tr>
<td></td>
<td>4.0.1r2</td>
<td>2010-06-17</td>
<td>Fix caption for CON/ADV/BI-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fixed wording in CON/SLA/BV-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fixed MSC for SEC/MAS/BV-01</td>
</tr>
<tr>
<td>1</td>
<td>4.0.1</td>
<td>2010-06-24</td>
<td>Prepare for publication</td>
</tr>
<tr>
<td></td>
<td>4.0.2r0-r3</td>
<td>2010-11-24</td>
<td>TSE 3517: TP/SEC/MAS/BI-07-C: update purpose; new test cases TP/SEC/MAS/BV-04-C,</td>
</tr>
</tbody>
</table>

**Publication Number:** 1034 of 1092
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2011-02-07</td>
<td>TP/SEC/MAS/BV-05-TP/SEC/MAS/BI-08-C, TP/SEC/MAS/BI-09-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3634 (supersedes TSE 3641):</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- updates to TP/TIM/SLA/BV-02-C, TP/TIM/SLA/BV-04-C, TP/TIM/BV-05-C,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- new test cases TP/TIM/ADV/BV-01-C, TP/TIM/ADV/BV-02-C, TP/TIM/ADV/BV-06-C,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/ADV/BV-07-C, TSE 3761: TP/CON/SLA/BV-02-C, TP/CON/MAS/BV-02, TP/CON/MAS/BV-22-C;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>updated MSCs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3927: TP/DDI/ADV/BV-15-18, TP/DDI/SCN/BV-05, TP/DDI/SCN/BV-12-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3939: TP/CON/SLA/BV-21-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/CON/ADV/BV-04-C: edit Pass/Fail verdicts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Input corrections per AT4Wireless review for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3517</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SEC/MAS/BI-07 verdicts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TPSEC/MAS/BI-09 MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SEC/MAS/BV-04 test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SEC/MAS/BV-05 pagination</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TCMT:- consolidate rows</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3634</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rev History for TSE 3641</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/ADV/BV-01: MSC caption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/ADV/BV-02: MSC caption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/SLA/BV-04: Init Condition SCA value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/SLA/BV-05: Init Condition SCA value, MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/SLA/BV-06: MSC caption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/TIM/SLA/BV-07: MSC caption (did not see Notes formatting issue)</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TSE 3297: Note that this is 3927 and is correct as is.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TSE 4098: TP/DDI/ADV/BV-02; update verdicts (remove 90%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TSE 4111:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TP/DDI/ADV/BV-07, update MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TP/DDI/ADB/BV-09; update MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TP/DDI/ADV/BV-11: update MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TSE 3634: TP/TIM/SLA/BV-05: Changed Initial Condition value from -20ppm to +20ppm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-07-18</td>
<td>TP/CON/ADV/BV-04- Fixed graphic to fit on one page</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-11-08</td>
<td>TSE 4351: new test case TP/SEC/SLA/BV-05-C, update TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2011-11-08</td>
<td>TSE 4459: TP/CON/MAS/BV-02-C: Change Test Procedure, Pass verdict.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>4.0.3r1</td>
<td>2011-11-09</td>
<td>TSE 3666: TP/CON/SLA/BI-05-C per Antonio’s instructions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4349: TP/CON/MAS/BV-19-C. Corrected placement of Optional box.</td>
</tr>
<tr>
<td>4.0.4r0</td>
<td></td>
<td>2012-05-18</td>
<td>TSE 4631: TP/DDI/ADV/BV-06-C, TP/DDI/ADV/BV-07-C, TP/DDI/ADV/BV-09-C: MSC corrections (HCI_<em>LE</em>_.....).</td>
</tr>
<tr>
<td>4.0.4r1</td>
<td></td>
<td>2012-07-05</td>
<td>TSE 4814: TP/CON/ADV/BI-01-C MSC and verdict updates</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4816: TP/CON/INI/BI-02-C MSC and verdict updates</td>
</tr>
<tr>
<td>4</td>
<td>4.0.4</td>
<td>2012-07-24</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>4.0.5r1</td>
<td></td>
<td>2013-05-31</td>
<td>TSE 4975: Updated test procedure of TP/SEC/MAS/BI-08-C to change to the correct error code.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5082: Updated test procedure and MSC for TP/DDI/ADV/BV-11-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5143: Updated test procedure and MSC for TP/SEC/MAS/BV-03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5144: Updated test case description, MSC, and Step 5 of the test procedure for TP/SEC/MAS/BI-08-C.</td>
</tr>
<tr>
<td>4.0.5r2</td>
<td></td>
<td>2013-06-03</td>
<td>BTI Review, Comments from Miles.</td>
</tr>
<tr>
<td>5</td>
<td>4.0.5</td>
<td>2013-07-02</td>
<td>Prepare for Publication</td>
</tr>
<tr>
<td>4.0.6rT, 4.0.6rTr3, 4.0.4rTr4</td>
<td></td>
<td>2013-07-07, 2013-09-25</td>
<td>Template Conversion:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Fail Verdicts Removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- New Pass/Fail Verdict Criteria section added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Removal of sections marked “N/A”</td>
</tr>
<tr>
<td>4.1.0r01</td>
<td></td>
<td>2013-09-25</td>
<td>Low Duty Cycle Directed Advertising CR</td>
</tr>
<tr>
<td>4.1.0r02</td>
<td></td>
<td>2013-10-10</td>
<td>LE Ping CR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5388: In Figure 4.1 updated the wording to read “the BER on the RF is less than 0.1%”.</td>
</tr>
<tr>
<td>4.1.0r03</td>
<td></td>
<td>2013-10-14</td>
<td>LE Link Layer Topology CR</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>4.1.0</td>
<td>2013-12-03</td>
<td>Prepare for Publication</td>
</tr>
<tr>
<td></td>
<td>4.1.1r01</td>
<td>2014-11-05</td>
<td>BTI Review, Dave, capitalized “Upper Tester” in new test cases, TP/SEC/MAS/BV-12-C and TP/SEC/MAS/BV-13-C.</td>
</tr>
<tr>
<td></td>
<td>4.2.0r00</td>
<td>2014-11-17</td>
<td>Integrated changes from Section 6 of Core_LE_Data_Length_Extensions_TEST.CRr01 and Sections 1.7 &amp; 1.9 of Core_Enhanced_Privacy_1_2.TS.CR.R05</td>
</tr>
<tr>
<td></td>
<td>4.2.0r01</td>
<td>2014-11-20</td>
<td>Integrated Reviews from Mayank and Rasmus</td>
</tr>
<tr>
<td></td>
<td>4.2.0r02</td>
<td>2014-11-24</td>
<td>Added captions to new MSCs. Minor editorial fixes</td>
</tr>
<tr>
<td></td>
<td>4.2.0r03</td>
<td>2014-11-25</td>
<td>Addressed outstanding comments from Mayank in 4.3.4 &amp; 4.3.5.</td>
</tr>
<tr>
<td></td>
<td>4.2.0r04</td>
<td>2014-11-25</td>
<td>Review by Dave. Minor editorial changes</td>
</tr>
<tr>
<td>7</td>
<td>4.2.0</td>
<td>2014-12-04</td>
<td>Prepare for TCRL 2014-2 publication</td>
</tr>
<tr>
<td></td>
<td>4.2.1r00</td>
<td>2015-05-06</td>
<td>TSE 6122: Added new test case TP/SEC/MAS/BV-14-C for unexpected Data Channel PDUs during encryption start (IUT Master). Updated TCMT accordingly. TSE 6181: Updated TP/DDI/SCN/BV-14-C to change random address to resolvable private address. TSR 6174: Updated TP/DDI/SCN/BV-14-C to correct Scan Filter Policy value from 0x04 to 0x02</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6292: Updated MSC for TP/DDI/SCN/BV-15-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6033: Editorial correction to test procedure for TP/CON/SLA/BV-10-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6034: Updated MSC and step 12 of test procedure for TP/DDI/ADV/BV-19-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6084: Changed Data length from FF to FFFFFFFF in TP/CON/MAS/BV-02-C &amp; TP/CON/MAS/BV-22-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6248: Corrected TCMT mapping for TP/SEC/SCN/BV-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6134: Replaced MSCs for TP/CON/INI/BV-01-C, TP/CON/INI/BV-02-C, TP/CON/INI/BV-04-C, TP/CON/INI/BI-02-C, TP/ENC/INI/BI-01-C to allow jitter in the first packet of a connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6154: Updated MSC and test procedure in TP/SEC/SLA/BI-05-C to allow option to send LL_ENC_RSP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6260: Updated item mapping in TCMT to remove redundant mapping to Table 1 items.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6213: Clarified tests by replacing “empty data packet” and similar terms with “Correctly formatted LL Data Channel PDU” in text and MSCs. Test groups affected: TP/CON/ADV &amp; TP/CON/INI</td>
</tr>
<tr>
<td></td>
<td>4.2.1r01</td>
<td>2015-05-18</td>
<td>Following review on BTI call May 18 2015, extended changes from TSE 6213 through the entire CON/ADV &amp; CON/INI test groups.</td>
</tr>
<tr>
<td></td>
<td>4.2.1r02</td>
<td>2015-06-15</td>
<td>TSE 6149: For tests TP/CON/SLA/BV-19-C, TP/CON/SLA/BI-05-C, and TP/CON/MAS/BV-20-C – updated MSCs and test procedures to allow for cases where the LL autonomously initiates the LL_VERSION_IND transaction.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2015-07-14</td>
<td>Editorial updates to correct specification references</td>
</tr>
<tr>
<td></td>
<td>4.2.2r00</td>
<td>2015-10-13</td>
<td>Prepared for publication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6491 &amp; 6655: Corrected terminology in MSCs from HCl_LE_Set_Random_Private_Address_Timeout to HCl_LE_Set_Resolvable_Private_Address_Timeout. Affects TP/CON/INI/BV-08-C,TP/CON/INI/BV-09-C,TP/CON/INI/BV-10-C,TP/CON/INI/BV-11-C,TP/CON/INI/BV-12-C,TP/DDI/SCN/BV-13-</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2015-12-22</td>
<td>Prepared for TCRL 2015-2 publication.</td>
</tr>
<tr>
<td>9</td>
<td>4.2.2</td>
<td>2016-02-12</td>
<td>TSE 6737: Unnecessary step deleted from test procedure of test case TP/SEC/ADV/BV-03-C.</td>
</tr>
<tr>
<td></td>
<td>4.2.3r00</td>
<td>2016-03-02</td>
<td>TSE 6865: Global edit. Changed ADV_DISCOVER_IND to ADV_SCAN_IND in body text and MSC (Figure 4.33).</td>
</tr>
<tr>
<td></td>
<td>4.2.3r01</td>
<td>2016-03-10</td>
<td>TSE 6652: Updated test case TP/ENC/SLA/BI-01-C: State, Test Procedure, Pass Verdict, and Notes.</td>
</tr>
<tr>
<td></td>
<td>4.2.3r02</td>
<td>2016-04-06</td>
<td>TSE 6103: Updated MSCs and test procedures for test cases TP/CON/SLA/BV-22-C, TP/CON/SLA/BV-23-C, TP/CON/MAS/BV-13-C, and TP/SEC/MAS/BV-11-C. Deleted entire section (4.3.4.39) for test case TP/CON/SLA/BI-06-C. Deleted row from TCMT for test case TP/CON/SLA/BI-06-C. Deleted 6/14 from TCMT for test case TP/CON/SLA/BI-05-C.</td>
</tr>
<tr>
<td></td>
<td>4.2.3r04</td>
<td>2016-04-12</td>
<td>TSE 6623: Deleted &quot;but which is generated using the IRK distributed to the IUT&quot; from Test Procedure step 3 for test case TP/DDI/SCN/BV-14-C. TSE 6656: Updated MSC per BTI discussion and TSE comment #22926. Deleted steps 9–12 in the Test Procedure and the first sentence of the Pass Verdict for test case TP/CON/INI/BV-08-C. TSE 6941: Updated text for Section 4.1.5.7, Common Test Procedure Steps. Extensive editorial and technical updates to Section 4.1.5.7.3, Optional Test Steps. TSE 6970: Updated MSC and figure caption. Updated step 2 of the Test Procedure and the Pass Verdict for test cases TP/SEC/SLA/BV-07-C and TP/SEC/MAS/BV-07-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6977: Updated first paragraph and Step 15 of Test Procedure for test case TP/DDI/ADV/BV-03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6983: Test case TP/SEC/ADV/BV-01-C updated: Section heading, test case description, Reference, Initial Condition State, Test Procedure (including MSC), and Pass Verdict. Updated TCMT, Feature name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6992: Optional HCI event added to MSC for test case TP/SEC/MAS/BV-14-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6993: Typo corrected. “Step 6” changed to “Step 4.”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7015: Deleted second sentence of Pass Verdict of test case TP/DDI/ADV/BV-19-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7016: Updated MSC and Pass Verdict for test cases TP/CON/SLA/BV-39-C and TP/CON/MAS/BV-40-C.</td>
</tr>
<tr>
<td></td>
<td>4.2.3r05</td>
<td>2016-04-25</td>
<td>Updated MSCs for test cases TP/CON/SLA/BV-23-C, TP/CON/MAS/BV-13-C, and TP/SEC/MAS/BV-11-C: “...FEATURE_REQ” and “…RSP” changed from solid to dashed arrows.</td>
</tr>
<tr>
<td></td>
<td>4.2.3</td>
<td>2016-07-13</td>
<td>Prepared for TCRL 2016-1 publication</td>
</tr>
<tr>
<td></td>
<td>5.0.0r00</td>
<td>2016-07-07</td>
<td>Integrated changes for Core Specification 5.0 release</td>
</tr>
<tr>
<td></td>
<td>5.0.0r01</td>
<td>2016-06-06</td>
<td>Test Spec Issue 7186: Updates test cases TP/CON/SLA/BV-45-C, TP/CON/SLA/BV-51-C, TP/CON/SLA/BI-09-C, and TP/CON/MAS/BV-46-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>5.0.0r04</td>
<td>2016-08-10</td>
<td>Issue 7487: Deleted duplicate test cases TP/FRH/SLA/BV-04-C and TP/FRH/SLA/BV-05-C.</td>
</tr>
<tr>
<td></td>
<td>5.0.0r05</td>
<td>2016-08-12</td>
<td>Issue 7465: Updated MSC, Pass Verdict, and Test Procedure (step 3) for test case TP/CON/SLA/BV-44-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0.0r07</td>
<td>Issue 7602: Added “Skip to step 11” to step 9 of test case TP/DDI/ADV/BV-24-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2016-09-27</td>
<td>Issue 7615: Changed property value for Round 1 from &quot;0x0040&quot; to &quot;0x0000&quot; in table in test case TP/DDI/ADV/BV-24-C. Changed max length of data for Round 2 from &quot;252&quot; to &quot;31&quot; in table in test case TP/DDI/ADV/BV-24-C and TP/DDI/ADV/BV-25-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7618: Changed “AUX_ADV_IND” to “ADV_EXT_IND” in step 7 of test case TP/DDI/ADV/BV-24-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7619: Updated test case TP/DDI/ADV/BV-24-C: Changed &quot;IUT stops advertising&quot; to &quot;IUT does not start any additional advertising events&quot; in step 9. Changed &quot;IUT stops advertising within the expected duration&quot; to &quot;IUT does not start any new advertising events after the time specified for Duration has elapsed&quot; in Pass Verdict. Added Notes section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7622: Updated step 1 in test cases TP/DDI/ADV/BV-19-C – 21-C: Changed “Own_Address_Type[0]” to “Own_Address_Type shall be.” Changed “Scanning_Filter_Policy[0]” to “Scanning_Filter_Policy shall be.” Updated step 1 in test cases TP/DDI/ADV/BV-23-C – 25-C: Changed “Own_Address_Type[i]” to “Own_Address_Type shall be.” Changed “Scanning_Filter_Policy[i]” to “Scanning_Filter_Policy shall be.” Updated step 1 in test case TP/CON/INI/BV-13-C: Updated step 1 parameters. Changed “Initiating_Filter_Policy[0]” to “Initiating_Filter_Policy shall be.” Changed “Own_Address_Type[0]” to “Own_Address_Type shall be.”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7623: Updated step 3 of test case TP/DDI/SCN/BV-23-C - 25-C. Updated table in test case TP/DDI/SCN/BV-23-C and 24-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7572: Updated Pass Verdict for test case TP/DDI/ADV/BV-01-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7624: Added 2 steps to the section Common Initial and Final Conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7627: Updated Pass Verdict for test case TP/DDI/ADV/BV-24-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7681: Moved “ChSel” bit in sections DDI &gt; Common PDU Contents and CON &gt; Common PDU Contents.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7682: Added item HCI 13/13 and feature name “LE Set Minimum Number Of Used Channels Command” to test case TP/FRH/SLA/BV-03-C in TCMT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7684: Updated test case TP/TIM/SCN/BV-02-C: Added text to introduction, updated MSC, deleted table and references to table in the Test Procedure steps, added Notes. Added new section for test case TP/TIM/SCN/BV-03-C. Added new TCMT entry for test case TP/TIM/SCN/BV-03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7705: Updated tests to allow enabling/disabling multiple sets for LE Set Extended Advertising Enable command. Replaced “a specified advertising set and duration” with “Duration[0]” for test case TP/DDI/ADV/BV-22-C. Changed “Duration” to “Duration[0]” for test cases TP/DDI/ADV/BV-24-C – 28-C, TP/DDI/ADV/BV-34-C, TP/CON/ADV/BV-05-C &amp; 06-C, and TP/TIM/ADV/BV-03-C &amp; 04-C. Updated steps 6 and 11 for test cases TP/DDI/ADV/BV-29-C – 32-C. Updated step 4 for test case TP/DDI/ADV/BV-33-C. Updated both MSCs for test case TP/DDI/ADV/BV-31-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7718: Updated steps 10 and 12 of test case TP/DDI/ADV/BV-33-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7721: Added parameters for Direct Address Type and Direct Address in LE Extended Advertising Report for test cases TP/DDI/SCN/BV-19-C and 20-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 7722: Updated test case TP/DDI/SCN/BV-19-C: Added scan timeout coverage. Updated step 2. Added new step 5. Updated steps 8 and 9</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
|                    | 5.0.0r08         | 2016-09-29| accordingly. Added new column (LE Set Extended Scan Enable) and new row (16) to table. Added new condition to Pass Verdict.  
Issue 7728: Deleted test case TP/FRH/SLA/BV-03-C and its corresponding TCMT reference. |
|                    | 5.0.0r09         | 2016-10-11| TSE 6899: Added new section “Data Fragmentation Over HCI”. Added reference to Section 4.1.7 in Pass verdict for test cases TP/CON/SLA/BV-04-C – 06-C, TP/CON/MAS/BV-03-C – 05-C, and TP/SEC/SLA/BV-05-C. Modified test procedure for test cases TP/CON/MAS/BV-03-C – 05-C and TP/CON/SLA/BV-05-C. Global edits: 1) Changed “1000 data packets” to “100 data packets” (i.e. 100 data packets = 1000 data bytes). 2) Deleted all references to “LL Defragmentation”  
TSE 7138: Standby State Preamble Steps updated: five MSCs replaced with one.  
TSE 7150: Corrected numbering in steps 4 and 9 for test case TP/SEC/SLA/BI-01-C.  
TSE 7242: Updated MSC and modified Test Procedure and Pass Verdict for test case TP/CON/SLA/BV-26-C. |
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7291: Added “LL 4/6” to mapping for test case TP/DDI/SCN/BV-14-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7438: Updated MSCs and Test Procedures (&quot;connection termination&quot; step) for TP/CON/INI/BV-09-C and TP/SEC/ADV/BV-07-C – 13-C. Updated last step of Test Procedure for TP/CON/INI/BV-10-C – 12-C. Updated MSC for TP/SEC/ADV/BV-06-C. Deleted second to last step of Test Procedure for TP/SEC/ADV/BV-08-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7463: Updated Test Procedures steps for TP/DDI/SCN/BV-16-C – 18-C. Added new step 1 to Test Procedure. References to other steps in same Test Procedure updated accordingly. Updated MSC for TP/DDI/SCN/BV-16-C: Modified RPA Timeout position and start/end points. Updated MSC for TP/DDI/SCN/BV-18-C: Changed &quot;(Random_Address)&quot; in first line to &quot;(Random_Static_Address)&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7575: Delete “Backoff Procedure” from Test Suite Structure figure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7582: Global edit. Changed “advertisement packets” to “advertising packets.”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7596: Updated MSC and Test Procedure for TP/SEC/ADV/BV-04-C and TP/SEC/ADV/BV-06-C. TP/SEC/ADV/BV-06-C MSC updates: Deleted first two commands (&quot;HCI_LE_Add_Device_To_Resolving_List&quot; and &quot;HCI_LE_Set_Address_Resolution_Enable&quot;) from Upper Tester to IUT and responses (HCI_Command_Complete_Event) from IUT to Upper Tester. Changed &quot;Executed entire Procedure 4 times&quot; to &quot;...3 times&quot; and deleted &quot;4. A Resolvable Private Address&quot; from configuration options.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>TSE 6693: Global edit. Modified PDUs in applicable body text and MSCs: Changed &quot;LL_CONNECTION_UPDATE_REQ&quot; to &quot;LL_CONNECTION_UPDATE_IND&quot;; &quot;LL_CHANNEL_MAP_REQ&quot; to &quot;LL_CHANNEL_MAP_IND&quot;; &quot;LL_REJECT_IND_EXT&quot; to &quot;LL_REJECT_EXT_IND&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.0.0r11</td>
<td>2016-11-10</td>
<td>Issue 8002: In the SCN test procedure, update Step 9 to identify received events as HCI LE Periodic Report events. TSE 7812: Updated Test Procedure step 6 for TP/DD/SCI/BN/BV-18-C. Added new test case TP/DD/SCI/BI-03-C. Updated TCMT with test case TP/DD/SCI/BN/BI-03-C. Issue 8023: Updated TP/DD/ADV/BV-28-C, Test Procedure, Step 7. In Test Procedure table, changed the Round 2 data length in &quot;HCI_LE_Set_Extended_Advertising_Data (Step 4)&quot; from &quot;0&quot; to &quot;1&quot;. Updated figure in TP/DD/ADV/BV-28-C. Added new Step 2 to the Test Procedure in TP/DD/ADV/BV-29-C. Issue 8065: Updated tests for periodic advertising scanning - TP/DD/SCI/BN/BV-21-C [Extended Scanning, Periodic Advertising Reception] and TP/DD/SCI/BN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)] - to set Unused to 0xFF for</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>5.0.0</td>
<td>2016-12-13</td>
<td>Approved by BTI. Prepared for TCRL 2016-2 publication.</td>
</tr>
<tr>
<td>11</td>
<td>5.0.1r00</td>
<td>2017-03-01</td>
<td>TSE 8656: Changed &quot;HCI_LE_Encryption_Change&quot; to &quot;HCI_Encryption_Change&quot; in test cases TP/SEC/MAS/BV-01 and TP/SEC/MAS/BV-04.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7854: Changed TP/SEC/ADV/BV-03-C Initial condition from &quot;Non-connectable Advertising (selected Adv Interval Min, selected Adv Interval Max, supported type of own address, selected advertising channel map)&quot; to “Device Address Set (supported type of address, any address)”. Added “according to IUT address set in initial condition” to the end of step 3 test procedure for figure 4.284 in TP/SEC/ADV/BV-03-C. Not part of TSE but renumbered test procedure steps for figure 4.284 in TP/SEC/ADV/BV-03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7843: added an HCI_LE_Set_Resolvable_Private_Address_Timeout to the MSC for TP/SEC/ADV/BV-13-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7852: Added new steps 9-11 to TP/SEC/ADV/BV-12-C and updated MSC accordingly. Added HCI Disconnect in final sequence of MSC for TP/SEC/ADV/BV-13-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7856: In the TP/CON/MAS/BV-07-C MSC, moved HCI_LE_Connection_Update_Complete_Event to the end of the sequence and fixed the typo in &quot;<em>Complete</em>&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 7860: clarified Own_Address_Type requirements in TP/SEC/ADV/BV-13-C and added RPA requirement in initA address to pass verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8158: expanded step 1 of TP/CON/MAS/BV-43-C to account for the full scope of required and allowed behavior. Updated MSC accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8304: added &quot;or until IUT sends a SCN_RSP&quot; to step 6 in parts a and b of test procedure of TP/DDI/ADV/BV-05-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8293: Updates to test procedures and MSCs to clarify timing and deviation in TP/CON/INI/BV-01-C, TP/CON/INI/BV-13-C, TP/CON/INI/BV-14-C, TP/CON/INI/BV-15-C, TP/CON/INI/BV-16-C, TP/CON/INI/BV-17-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8327: TP/CON/INI/BV-20-C - corrected step reference to step 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8345: In MSC for TP/CON/SLA/BV-57-C, changed LL_PHY_REQ Both Directions = 1Ms/s to LL_PHY_REQ Both Directions = LE Coded PHY</td>
</tr>
<tr>
<td>5.0.1r01</td>
<td>2017-03-08</td>
<td></td>
<td>TSE 8293: Updates to test procedures and MSCs to clarify timing and deviation in TP/CON/INI/BV-01-C, TP/CON/INI/BV-13-C, TP/CON/INI/BV-14-C, TP/CON/INI/BV-15-C, TP/CON/INI/BV-16-C, TP/CON/INI/BV-17-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8327: TP/CON/INI/BV-20-C - corrected step reference to step 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8345: In MSC for TP/CON/SLA/BV-57-C, changed LL_PHY_REQ Both Directions = 1Ms/s to LL_PHY_REQ Both Directions = LE Coded PHY</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>TSE 8348: Fixed difference between MSC and Test Procedure text in TP/CON/SLA/BV-57-C and TP/CON/MAS/BV-55-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8369: added MSC figure to TP/CON/SLA/BV-42-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8376: Corrected “expects with” to “expects” in TP/CON/MAS/BV-50-C, TP/CON/MAS/BV-54-C, and TP/CON/MAS/BV-55-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8377: Corrected typo in Step 8 of TP/CON/SLA/BV-58-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8378: In TP/CON/SLA/BV-54-C MSC, updated the first “2Ms/s to “LE Coded PHY”. In TP/CON/SLA/BV-57-C MSC part B, “1Ms/s” updated to “LE Coded PHY”.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8379: Deleted HCI_LE_Set_PHY and HCI_Command_Status_Event at start of Part B of MSC in TP/CON/MAS/BV-54-C and TP/CON/MAS/BV-55-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8380: Removed unnecessary HCI_LE_Set_PHY from TP/CON/SLA/BV-51-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8383: Corrected formatting error in “MAS” heading for Section 4.3.5.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8388: Corrected typo in TP/DDI/ADV/BV-27-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8390: Deleted “On Round 1 only…” in step 1 of TP/DDI/ADV/BV-27-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8413: moved “…that was used in the SCAN_REQ packet” from step 6 to step 7 in TP/DDI/SCN/BV-18-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8416: Replaced 0xFF with 0x00 for Unused value in TP/DDI/SCN/BV-21-C, TP/DDI/SCN/BV-25-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 8468: In Section 4.1.6, added the recommendation that the control of the IUT is arranged for testing purposes so that Inconclusive verdicts are the exceptions rather than the rule. TSE 8587: In TP/DDI/ADV/BV-25-C, corrected scan data length in rounds 1 and 9 from 0 to 1.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 5.0.1r02           |                  | 2017-03-15| TSE 8384: In TP/CON/MAS/BV-51-C: Deleted ”LE Coded“ from the title. In the MSC, changed "0x02" to "0x02 or 0x04" in two places (first step). In step 1, changed: 1) "zero and" to "zero," and 2) "prefer a PHY" to "prefer a single supported PHY". In steps 3a/b, changed 1) "the opposite bit in" to "a different bit in the"; 2) change "both bits" to "more than one bit". Appended to step 3a: "The bit set must correspond to a PHY that the IUT supports." In TCMT, remapped TP/CON/MAS/BV-51-C to feature "PHY Update Procedure (Master)."
|                    |                  |            | TSE 8389: In TP/DDI/ADV/BV-27-C: Modified step 7: Deleted “On Round 1, or if the Advertising Data ID has changed since the last round (N-1),” and the final paragraph, "If the Advertising Data ID is the same as the last round, skip to Step 10." Added new step to Test Procedure. In step 10, changed “6–8” to “6–9”. In the Pass Verdict: 1) Deleted from the third bullet: "If advertising data was specified for the round"; 2) added fourth bullet: "The Advertising Data ID changes whenever the data changes.” |
| 5.0.1r03           |                  | 2017-03-16| TSE 8222: Various fixes in TP/CON/ADV/BV-05-C and 06-C to correct transition from Initiating State to Connection State as well as other procedural corrections
|                    |                  |            | TSE 7859: TP/SEC/ADV/BV-05-C: In the MSC, for HCI_LE_Set_Advertising_Parameters, add Adv_filter_policy =0x01; TP/SEC/ADV/BV-07-C: In the MSC, for HCI_LE_Set_Advertising_Parameters, add Adv_filter_policy =0x02
|                    |                  |            | TSE 7879: Removed step 7, added text to step 3 and updated numbering in step 15 of TP/CON/INI/BV-12-C to clarify IUT address type.
|                    |                  |            | TSE 8305: Updates to interval length in test procedure and MSC for TP/DDI/ADV/BV-11-C.
|                    |                  |            | TSE 8320: Corrected the Advertising_SID value in step 1 of TP/DDI/ADV/BV-35-C.
|                    |                  |            | TSE 8335: LeSetPhy Param ALLPHYS corrected to 0x03 instead of 0x05 in TP/CON/SLA/BV-54-C.
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5.0.1r04</td>
<td>2017-04-18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0.1r05</td>
<td>2017-05-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>procedures according to the new numbering. Removed “Repeat steps 1-7 for each Round shown in Table 4.16.” from the whole test procedure. Updated test procedure step 9 and 10 to include “only rounds 5-8 in Table 4.16 are executed, in Step 2” and removed “and in Step 8 only rounds 5-10 in Table 4.16 are executed”. Changed the scan data for Table 4.16 from None to “1” for Round 5, 6 and 7. Changed from Step 3 to “Step 4” in pass verdict. TSE 8289: Removed 2 text boxes “It is checked that no packets....”. All instances of LL_CONNECTION_UPDATE_REQ are replaced by LL_CONNECTION_UPDATE_IND. Fixed spelling of “transmission” from MSC for TP/TIM/SLA/BV-01-C. Removed text box “It is checked that no packets....” from MSC for TP/TIM/SLA/BV-02-C and TP/TIM/SLA/BV-03-C. TSE 8323: Revised the first part of step 11 test procedure and replaced &quot;PHY bit in the LL_PHY_RSP&quot; with &quot;bit in the TX_PHYS field&quot; for the last inconclusive verdict for TP/CON/SLA/BV-53-C. Revised the first part of step 8 test procedure and replaced &quot;allow the Lower Tester to select the LE Coded PHY&quot; with &quot;set the LE Coded PHY bit in the TX_PHYS field&quot; for the last inconclusive verdict for TP/CON/SLA/BV-59-C. TSE 8362: Added dotted arrow for AUX-ADV_IND to MSC Figure 4.63 and updated step 4 of test procedure for TP/DDI/ADV/BV-34-C. TSE 8579: Various fixes to TP/CON/MAS/BV-50-C: Removed steps 1 and 2 from the test procedure of TP/CON/MAS/BV-50-C and renumbered steps in test procedure, pass verdict, and inconclusive verdict to match new numbering. Updated MSC Figure 4.235 to reflect removal of step 1 and 2 from test procedure. TSE 8584: Corrected “andif” to “and if” in Pass/Inconclusive/Fail Verdict Conventions section. TSE 8586: Replaced line 3 of pass verdict with &quot;The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms,” for TP/DDI/ADV/BV-15-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
</tbody>
</table>
|                    | 5.0.1r06         | 2017-05-15 | TSE 8860: Removed (Identity Address) from SCAN_RSP in MSC Figure 4.299 and changed part of pass verdict from “Lower Tester’s device identity address” to “the advertiser RPA” for TP/SEC/ADV/BV-18-C.  
TSE 8895: Changed title of TP/SEC/MAS/BV-13-C from “FEATURES” to “FEATURE”.  
TSE 8844: Added clarifying text about own address type “(0×02 or 0×03) to initial condition of TP/DDI/ADV/BV-09-C, TP/DDI/SCN/BV-13-C, TP/DDI/SCN/BV-14-C, TP/DDI/SCN/BV-26-C.  
TSE 8681: Renumbered test procedure steps to start at 1 for TP/SEC/ADV/BV-03-C.  
TSE 8548: Updated MSC Figure 4.296 with “Advertising_Filter_Policy” and added text “Advertising Filter Policy shall be set to 0x01” to step 3 of the test procedure for TP/SEC/ADV/BV-15-C.  
TSE 9040: Replaced 2µs with 1.5µs in MSC and in step 2 of the test procedure, and added notes to the pass verdict for TP/TIM/MAS/BV-01-C, TP/TIM/MAS/BV-02-C.  
TSE 8523: Various fixes to TP/DDI/ADV/BV-33-C: Updated MSC to show the enable command being executed only once, added text to allow for an inconclusive verdict when a set cannot be created to step 1 and 2 of the test procedure, Updated step 5 to specify that the Lower Tester scans for each adv. PHY supported by the IUT. Clarified need to be repeated for each “first set” to steps 9-11 in the test procedure, corrected the pass verdict to specify conditions for each adv. set, rather than for each |
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8530: Clarified order of procedures with changes to steps 7-8 of TP/CON/MAS/BV-28-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8532: added clarification of behavior based on Max_Extended_Advertising_Events parameter value via changes to test procedure steps, reference tables, and Pass verdict in TP/DDI/ADV/BV-24-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8666: Added missing step 2 to test cases TP/DDI/ADV/BV-07-C – 09-C and updated subsequent step references.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8568: Clarified advertising interval length in test procedure steps of TP/DDI/ADV/BV-24-C – 35-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8665: Added steps to TP/DDI/ADV/BV-29-C and 32-C to clarify disabling and clearing of advertising sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8706: Added clause regarding advertising channel use for the various PHY to step 1 of TP/ADV/DDI/BV-33-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8822: Extensive changes to test procedures in TP/FRH/MAS/BV-01-C and 02-C. Replaced MSCs. Added Inconclusive verdicts and updated Table 4.2 accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8662: Extensive changes through steps 5-15 to clarify TP/DDI/SCN/BV-25-C. Corresponding updates to MSC.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5.0.1r11</td>
<td>2017-06-08</td>
<td></td>
<td>Editorial changes per review comments received against several TSEs (added in real time on BTI call) REMOVE THIS REV HISTORY LINE BEFORE PUBLICATION</td>
</tr>
<tr>
<td>12</td>
<td>5.0.1</td>
<td>2017-07-05</td>
<td>Approved by BTI. Prepared for TCRL 2017-1 publication.</td>
</tr>
<tr>
<td>5.0.2r00</td>
<td>2017-07-20</td>
<td></td>
<td>TSE 8580: Replace &quot;Repeat for up to 20 advertising intervals [...]&quot; by &quot;Repeat for at least 20 advertising intervals [...]&quot; in the following test cases.  LL/DDI/SCN/BV-15-C, 16-C, 17-C, 18-C, 19-C, 20-C, 23-C, 24-C and LL/DDI/SCN/BV-03-C  TSE 8865: For test case LL/CON/MAS/BV-13-C: Change “The tester acts in the slave role [...]” to “The Lower Tester acts in the slave role [...]” In Test Procedure step 2, changed name to match 5.0 spec. In step 3, revised for clarity and revised incorrect reference of REQ_PDU. In step 4, deleted extra use of “event” and changed name to match 5.0 spec. Deleted step 5. Updated the Pass Verdict. Added Note to cover the name update. Revised MSC. Updated to HCI_LE_Read_Remote_Features to match BT 5.0</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Core Spec. Changed &quot;LL_SLAVE_FEATURE_REQ&quot; to &quot;LL_FEATURE_REQ&quot; in MSC note.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8865 continued: For test case LL/SEC/MAS/BV-11-C:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added LL_prefix to REJECT_EXT_IND.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated MSC to remove the incorrect use of &quot;LL_SLAVE_FEATURE_REQ&quot; by master.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In Test Procedure, deleted step 1. In step 2, added LL_prefix. In step 3, changed formatting to match PDU variable name. In step 4, changed to match the pass verdict (i.e. error code becomes 'status parameter' since the HCI event doesn't have a variable named error code.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated text in Pass Verdict.</td>
</tr>
<tr>
<td>5.0.2r01</td>
<td></td>
<td>2017-08-16</td>
<td>TSE 9600: Updated step number in test procedure for LL/CON/ADV/BV-06-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9401: For LL/CON/MAS/BV-55-C, removed optional PHY update procedure from MSC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9403: For LL/CON/ADV/BV-06-C, deleted step 14; reordered bullets of the Pass Verdict by step number; and added Fail Verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9431: For LL/CON/SLA/BV-14-C, changed all instances of &quot;Features_Req packet&quot; to &quot;LL_FEATURE_REQ PDU&quot; and &quot;Features_Rsp&quot; to &quot;LL_FEATURE_RSP&quot;. For LL/CON/MAS/BV-19-C, changed all instances of &quot;Features_Req packet&quot; to &quot;LL_FEATURE_REQ PDU&quot; and &quot;Features_Rsp packet&quot; to &quot;LL_FEATURE_RSP PDU&quot;. Revised initial condition for LL/CON/SLA/BV-14-C and revised test procedure and Figure 4.206 for LL/CON/MAS/BV-19-C.</td>
</tr>
<tr>
<td>5.0.2r02</td>
<td></td>
<td>2017-08-18</td>
<td>TSE 8723: For LL/DDI/SCN/BV-11-C, updated introduction, initial condition, prior steps, and pass verdict text; replaced figure.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 5.0.2r03           |                  | 2017-08-22 | TSE 9402: For LL/CON/ADV/BV-05-C, revised MSC boxes.  
TSE 9602: Revised test procedure values in LL/DDI/ADV/BV-26-C.  
TSE 9707: Deleted LL/CON/SLA/BV-41-C from the table in Pass/Inconclusive/Fail Verdict Conventions.  
| 5.0.2r04           |                  | 2017-08-23 | TSE 9666: For LL/DDI/ADV/BV-24-C ...27-C, and 29-C ...32-C, changed the minimum value of Maximum_Advertising_Data_Length from 0x00BF to 0x001F.                                                               |
| 5.0.2r05           |                  | 2017-08-28 | TSE 9379: Updated step 1 and Figure 4.191 for LL/CON/SLA/BI-09-C.  
TSE 9404: For LL/DDI/SCN/BV-11-C and LL/CON/INI/BV-07-C, revised text and removed references to advertising data ADV_DIRECT_IND packets.  
TSE 9573: Revised pass verdict text for LL/CON/MAS/BV-41-C.  
TSE 9476: Corrected mapping for LL/CON/SLA/BV-04-C ...06-C and LL/CON/MAS/BV-03-C ...05-C. |
| 5.0.2r06           |                  | 2017-09-01 | TSE 8746: Added new test cases LL/PAC/SLA/BI-01-C and LL/PAC/MAS/BI-01-C to the PAC section and the TCMT.                                                                                      |
| 5.0.2r07           |                  | 2017-09-15 | TSE 8797: Revised text and figures for test case LL/DDI/ADV/BV-33-C. Updated Test Procedure step 1 in LL/DDI/ADV/BV-33-C.  
<p>| 5.0.2r08           |                  | 2017-09-19 | AoA/AoD CR incorporation: Updated sections: Test Strategy; Pass/Inconclusive/Fail Verdict Conventions; TCMT; LL/DDI/SCN/BV-21-C, LL/DDI/SCN/BV-25-C; New sections: LL/DDI/ADV/BV-36-C –37-C, |
|                    |                  |            |                                                                                                                                                                                                          |</p>
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.0.2r11</td>
<td>2017-10-02</td>
<td>TSE 8671: Revised LL/DDI/SCN/BV-19-C test procedure and pass verdict text.</td>
</tr>
<tr>
<td></td>
<td>5.0.2r12</td>
<td>2017-10-09</td>
<td>TSE 9885: Corrected style of PDU error code field to “ErrorCode” (global change).</td>
</tr>
<tr>
<td></td>
<td>5.0.2r14</td>
<td>2017-10-11</td>
<td>TSE 9662: Revised inconclusive verdicts table and LL/DDI/ADV/BV-33-C text, initial condition, MSC, and test procedure. TSE 9393: Revised test cases: LL/DDI/ADV/BV-05-C and 17-C text, initial condition, test procedure,</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSC, and expected outcome; LL/DDI/ADV/BV-06-C text, initial condition, and test procedure; LL/DDI/ADV/BV-09-C test procedure and expected outcome.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8686: Revised LL/DDI/SCN/BI-03-C initial condition, MSC, test procedure, and expected outcome.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 8670: Revised LL/CON/SLA/BI-02-C and LL/CON/MAS/BI-02-C test procedure and expected outcome.</td>
</tr>
<tr>
<td></td>
<td>5.0.2r15</td>
<td>2017-10-12</td>
<td>TSE 8100: Revised the “INI” subsection in the “CON” section. Revised MSCs: LL/CON/INI/BV-01-C, 04-C, 08-C – 17-C, 20-C, and 21-C; and LL/CON/INI/BI-02-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9749: Revised LL/DDI/ADV/BV-06-C, 07-C, and 09-C test procedures and MSCs.</td>
</tr>
<tr>
<td></td>
<td>5.0.2r17</td>
<td>2017-10-18</td>
<td>TSE 9930: Revised LL/CON/SLA/BI-04-C text, initial condition, MSC, test procedure, and expected outcome.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9927: Added new subsection “Outstanding Commands Prior to Disconnection” to the “Test Cases (TC),” “Introduction,” section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9897: For LL/CON/SLA/BV-40-C: revised test procedure, deleted the “PDU payload contents for each case variation for LE 2M PHY supported and LE Coded PHY not supported” and “PDU payload contents for each case variation for LE Coded PHY supported and LE 2M PHY not supported” tables, and revised the “PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported” table. Updated table numbers.</td>
</tr>
<tr>
<td></td>
<td>5.0.2r18</td>
<td>2017-10-19</td>
<td>TSE 9898: Added new test case LL/DDI/ADV/BV-38-C and added to the TCMT.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>5.0.2</td>
<td>2017-12-07</td>
<td>Approved by BTI. Prepared for TCRL 2017-2 publication.</td>
</tr>
<tr>
<td>13</td>
<td>5.0.3r00–17</td>
<td>2018-01-05 – 2018-06-20</td>
<td>Template update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 10212: For LL/DDI/ADV/BV-36-C, 37-C: Added new sentence “Advertisements without data, along with chaining, are tested”; replaced MSC; revised Test Procedure; updated Pass Verdict. For LL/DDI/SCN/BV-29-C, 30-C: Replaced MSC; revised Test Procedure; updated Pass Verdict. For LL/DDI/SCN/BV-31-C, 32-C: updated Test Procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 10180: Added new item LL 11/TBD1 to LL/DDI/SCN/BV-35-C, 36-C to TCMT. Added new test cases LL/DDI/SCN/BV-35-C, 36-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>LL/CON/MAS/BV-65-C, 66-C: Replaced MSC; updated Test Procedure; updated Pass Verdict.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10232 (rating 1): Moved Section 4.3.2 (Common PDU Contents) before Section 4.3.1 (ADV).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10204 (rating 1): Corrected Test Procedure for LL/DDI/SCN/BV-34-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10355 (rating 1): Replaced MSC and revised Test Procedure for LL/DDI/SCN/BV-19-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10186 (rating 1): Added Test Purpose heading and replaced MSC for LL/SEC/ADV/BV-02-C - 03-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10167 (rating 3): Revised test procedure step 5 in LL/DDI/ADV/BV-32-C to reject HCI_LE_Set_Extended_Advertising_Data commands only if the Data Length for a given set is greater than zero.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10328 (rating 3): Replaced MSC and revised test procedure step 3 for LL/TIM/SCN/BV-01-C and 02-C to clarify use of T_MAFS.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 9975 (rating 3): Added an inconclusive verdict as a possible outcome to the expected outcome for LL/CON/SLA/BV-20-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 9896 (rating 3): Replaced test procedure steps 7 and 8 for LL/CON/MAS/BV-28-C to make message order more flexible.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 9907 (rating 3): Removed mandatory condition from test procedure step 3 in LL/CON/MAS/BV-33-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10076 (rating 3): In test purpose, changed &quot;REJECT_EXT_IND&quot; to &quot;LL_REJECT_IND or LL_REJECT_EXT_IND&quot;; in MSC, made LL_REJECT_IND (ErrorCode: 0x06) be an alternative to the LL_REJECT_EXT_IND; in step 5, allowed either PDU to be sent; in expected outcome pass verdict, allowed either PDU to be sent; in TCMT, removed the requirement for LL 9/4 for LL/SEC/SLA/BV-11-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSE 10151 (rating 3): Revised repetitions from 100 to 10 in MSC; revised test procedure step 9; and added a column for repeat count (step 9) for LL/DDI/ADV/BV-28-C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10160 (rating 3): Revised test procedure steps 1 and 2 for LL/DDI/ADV/BV-33-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10234 (rating 3): Replaced MSC, revised test procedure step 3, and deleted notes to unconditionally allow both 0x16 and 0x22 as valid error codes for HCI_Disconnect_Complete_Event in LL/CON/MAS/BI-02-C and LL/CON/SLA/BI-02-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10382 (rating 1): Made editorial changes to test purpose in LL/CON/SLA/BV-49-C, 50-C, 52-C, 53-C, 55-C to 59-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10405 (rating 4): Added references 11 and 12 to section 2.1 (References). Added new section 4.9 (Data Flow). Added test cases LL/DFL/MAS/BV-01-</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C, 02-C, LL/DFL/SLA/BV-01-C, and 02-C and their corresponding TCMT entries.</td>
</tr>
<tr>
<td></td>
<td>TSE 10362 (rating 3): Modified test procedure step 11 and expected outcome inconclusive verdict for LL/CON/SLA/BV-52-C and 53-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10288 (rating 3): Replaced MSC with &quot;(TODO)&quot; and modified figure title; test procedure steps 9, 12, 17, and 18; Table 4.20; and expected outcome pass verdict for LL/DDI/SCN/BV-21-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 9884 (rating 3): Revised MSC and test procedure for LL/SEC/MAS/BI-07-C and 09-C. Revised expected outcome pass verdict for LL/SEC/MAS/BI-09-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Issue 10377: Replaced the figure and updated the Test Procedure and Pass Verdict for test cases LL/CON/SLA/BV-64-C and LL/CON/MAS/BV-60-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 9744 (rating 3): Inserted a new step 2 in the LL/TIM/ADV/BV-03-C and 04-C test procedure and renumbered the remaining steps. Corrected the referenced step number in the Pass Verdict from 3 to 4. Expanded the MSC to show the HCI LE Set Extended Scan Response Data command exchange on IUT.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10354 (rating 1): Moved and renumbered LL/DDI/ADV/BV-38-C to LL/DDI/SCN/BV-37-C and revised its corresponding TCMT entry (this was taken care of by TSE 10209). LL/DDI/ADV/BV-38-C is no longer used in the TS.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10357 (rating 1): Globally replaced &quot;Periodic Scanning&quot; with &quot;Scanning for Periodic Advertising&quot; in the TCMT.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10517 (rating 2): Changed 4/7 to 4/6a in the TCMT for test case LL/DDI/SCN/BV-34-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10513 (rating 2): Changed 4/8 to 4/7 in the TCMT for test case LL/CON/INI/BV-13-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10499 (rating 3): Changed filter value to 0x01 in the Notes for test case LL/DDI/SCN/BI-01-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Changed specific device to all unknown devices in the Notes for test case LL/DDI/SCN/BI-02-C, and added &quot;OR All White Listed (policy for scanner)&quot; to the Initial Condition State.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10478 (rating 3): Simplified the test purpose for test case LL/CON/SLA/BI-09-C. Replaced the figure with a new figure that includes ALT1 and ALT2. Changed steps 5 and 6 in the test procedure to Alternative 1 and added Alternative 2. Added Alternative 1 and 2 to the Pass Verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10353 (rating 3): Added Command Complete events to steps 1, 2, 3, and 10 in the test procedure for test case LL/DDI/SCN/BV-34-C. Changed HCI_Command_Status to HCI_Command_Complete in steps 14 and 17 and in the MSC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10343 (rating 3): Inserted a new step 1 in the LL/DDI/SCN/BV-33-C test procedure and renumbered the remaining steps. Added HCI_LE_Add_Device_To_White_List to beginning of MSC with complementary HCI Command Complete event.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10016 (rating 3): Editorial fix for step 3 in the LL/CON/MAS/BV-41-C test procedure as it was starting at the wrong point. Added clarifying text to the LL/CON/MAS/BV-42-C test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9884 (rating 3): reincorporated, removed step 5 of the LL/SEC/MAS/BI-09-C test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9975 (rating 3): reincorporated, added Inconclusive Verdict to test case LL/CON/MAS/BV-21-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9991 (rating 2): reincorporated, added item LL 9/6 to test cases LL/CON/MAS/BV-53-C and LL/CON/MAS/BV-54-C in the TCMT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10288 (rating 3): reincorporated, fixed typo in steps 9 and 12 of the LL/DDI/SCN/BV-21-C test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10610 (rating 3): Revised ADV_EXT_IND and AUX_ADV_IND parameters and changed HCI_LE_SET_Random_Private_Address_Timeout to</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>HCI_LE_SET_Resolvable_Private_Address_Timeout in the MSC for test case LL/DDI/SCN/BV-33-C. Updated step 5 of the Test Procedure.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10371 (rating 3): Updated the MSC to make the LE_PHY_Update_Complete optional for test case LL/CON/SLA/BV-40-C. Revised test procedure steps 2 and 9. Changed step 4 to body text and renumbered the subsequent steps. Revised the pass verdict.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 9908 (rating 3): Updated MSC and test procedure steps 1 and 2 for test cases LL/PAC/SLA/BV-01-C and LL/PAC/MAS/BV-01-C to clarify the kind of control PDUs that are supposed to be responded with LL_UNKNOWN_RSP.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10506 (rating 4): Added new test cases LL/CON/ADV/BV-11-C and LL/CON/INI/BV-22-C, and added them to the TCMT.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10509 (rating 3): Revised MSC for test case LL/CON/SLA/BV-04-C and updated and renumbered test procedure steps.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10499 (rating 3): Replaced MSC for test case LL/DDI/SCN/BI-02-C (changed HCI_LE_Set_Scan_Parameters command's policy from 0x00 to 0x01.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10530 (rating 3): Replaced MSC to show 40 iterations for test cases LL/ENC/ADV/BI-01-C and 02-C; LL/ENC/SCN/BI-01-C and 02-C; LL/ENC/INI/BI-01-C; and LL/ENC/MAS/BI-01-C. Updated test procedure steps to clarify that the corrupted bit position is changing with each iteration. Doing 2 sequences of 20 iterations makes 40 total. Updated pass verdict and Notes.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10603 (rating 4): Added missing MSC and removed test procedure step 10 for test case LL/CON/MAS/BV-73-C. Added missing MSC and revised test procedure step 4 for test case LL/CON/MAS/BV-74-C. Added missing MSC for test cases LL/CON/MAS/BV-75-C to 80-C.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TSE 10389 (rating 3): Modified Round 2 and 3, added new Round 4 in the test procedures for test cases LL/CON/MAS/BV-57-C and LL/CON/SLA/BV-61-C.</strong></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>E10745: Deleted test cases LL/CON/SLA/BV-60-C to 62-C, 64-C to 69-C, and 71-C to 76-C; LL/CON/MAS/BV-56-C to 58-C, 60-C to 65-C, and 67-C to 68-C. Added new test cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoA]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Periodic]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, Timeout]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoA]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoD]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoD]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Unrequested Constant Tone Extension, IUT Receiving, AoA]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoA, Encrypted Connection]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 11105: In LL/DDI/ADV/BV-43-C, replaced AUX_SYNC_IND by &quot;periodic advertising train&quot;; made additional related clarifications to the test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 10859: LL/TIM/SLA/BV-08-C, 10-C; LL/TIM/MAS/BV-03-c – 07-C: Made corrections to</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>test cases to allow Sleep Clock Accuracy procedure to trigger. Removed last paragraph of Initial Condition. Replaced MSC. Edited test procedure steps.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 11130: Modified steps 4 and 5 and added a fail verdict to LL/CON/SLA/BV-106-C and LL/CON/MAS/BV-102-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 11122: Deleted test cases belonging to enhancements out of the final Madrid scope: DDI/ADV/BI-03-C and 04-C; DDI/ADV/BV-40-C through 42-C; DDI/ADV/BV-44-C. Updated test case mapping accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9909 (rating 3): Updated MSC and test procedure step 3 for test case LL/CON/SLA/BV-13-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10021 (rating 4): Added new test case LL/CON/MAS/BV-117-C, and updated TCMT and section 4.1.5.8 Pass/Inconclusive/Fail Verdict Conventions Table 4.2: Test cases with allowable Inconclusive Verdicts with new test case.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10345 (rating 2): Updated section title, test procedure, MSC, expected outcome, and pass verdict for test case LL/DDI/ADV/BV-24-C. Updated TCMT with test cases LL/DDI/ADV/BV-TBD1-C and TBD2-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10649 (rating 3): Updated initial condition for test case LL/DDI/SCN/BV-33-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10190 (rating 3): Updated test procedure steps and fixed punctuation in pass verdict for test case LL/DDI/ADV/BV-09-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10668 (rating 1): Updated test procedure step 11 and pass verdict for test case LL/DDI/ADV/BV-25-C [Extended Advertising, Scannable].</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10675 (rating 1): Removed &quot;non white&quot; listed device address from test procedure steps for test cases LL/DDI/ADV/BV-05-C, 06-C, and 17-C; LL/DDI/SCN/BV-02-C and 04-C; LL/CON/INI/BV-06-C and 07-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10697 (rating 3): Added a note at the end of the test procedure for test case LL/DDI/SCN/BV-19-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10702 (rating 1): Changed &quot;Upper Tester&quot; to &quot;Lower Tester&quot; in test procedure step 4 for test case LL/DDI/ADV/BV-06-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10713 (rating 1): Updated test procedure step 11 for test case LL/CON/MAS/BV-74-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10768 (rating 1): Updated MSC for test case LL/CON/SLA/BV-26-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10807 (rating 3): Added optional sequence of test steps &quot;LE PHY Update Procedure (LE Coded Switch) Optional Test Steps&quot; to the end of section 4.1.5.7.3 Optional Test Steps.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10809 (rating 1): Revised test procedure step 14 for test case LL/DDI/ADV/BV-05-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10812 (rating 3): Updated test procedure steps 10 and 13 for test case LL/DDI/SCN/BV-33-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10817 (rating 3): Updated test procedure and pass verdict for test case LL/SEC/MAS/BV-10-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10866 (rating 3): Updated test procedure step 1 for test case LL/CON/SLA/BV-18-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10989 (rating 1): Updated MSCs for test case LL/DDI/ADV/BV-30-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11050 (rating 1): Updated MSCs for test cases LL/DFL/SLA/BV-01-C and 02-C; and LL/DFL/MAS/BV-01-C and 02-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11088 (rating 3): Updated test procedure step 6 for test case LL/DDI/ADV/BV-32-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11101 (rating 1): Updated MSC and test procedure step 9 for test case LL/CON/MAS/BV-50-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10190 (rating 3): Updated test procedure steps per new CR for test case LL/DDI/ADV/BV-09-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10656 (rating 2): Updated test procedure for test case LL/CON/MAS/BV-18-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 10808 (rating 3): Updated MSCs, test procedure, and expected outcome for test cases LL/CON/SLA/BV-57-C and LL/CON/MAS/BV-55-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.1.0</td>
<td>Updated revision number from 5.0.4 to 5.1.0 to align with the adoption of Core Specification version 5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2018-11-13</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11104 (rating 4): Added new test case LL/SEC/ADV/BV-21-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>15</td>
<td>5.1.0</td>
<td>2018-12-07</td>
<td>Approved by BTI. Prepared for TCRL 2018-2 publication.</td>
</tr>
</tbody>
</table>
|                    | 5.1.1r00–r22     | 2019-03-20–2019-07-02 | TSE 11573 (rating 1): Updated test procedure step 12 for test case LL/DDI/ADV/BV-37-C so that the check should be that the CTEType is set to 0 (AoA Constant Tone Extension).  
TSE 11503 (rating 1): Updated test procedure step 2 for test cases LL/CON/SLA/BV-105-C and LL/CON/MAS/BV-101-C to clarify that connEventCount does not need to be an event that has occurred.  
TSE 11459 (rating 1): Updated “Extended Advertising, Scannable Test Cases” table for TP/DDI/ADV/BV-25-C to clarify wording in Test Case descriptions.  
TSE 11401 (rating 1): Updated test procedure step 1 for test cases LL/DFL/MAS/BV-01-C and LL/DFL/SLA/BV-01-C to change wording to send the rest in “one or more continue fragments.”  
TSE 11145 (rating 1): Updated all text instances of “HCI_LE_Set_Advertise_Enable” to “HCI_LE_Set_Advertising_Enable” in text and MSCs.  
TSE 10549 (rating 2): Updated test steps in test cases LL/CON/MAS/BV-73-C – -75-C.  
TSE 11165 (rating 2): Updated TCMT Item and Feature columns for test case LL/CON/INI/BV-13-C.  
TSE 11571 (rating 3): Modified Test Purpose text, updated MSC, updated Test Procedure step, and updated Pass Verdict text for test cases LL/DDI/ADV/BV-08-C and -18-C. Updated Test Purpose text for test cases LL/DDI/ADV/BV-09-C |
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>and LL/DDI/SCN/BV-02-C. Updated Pass Verdict text for test cases LL/CON/INI/BV-06-C and -07-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11553 (rating 3): Combined test cases LL/CON/MAS/BV-32-C and -33-C into one test procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11464 (rating 3): Updated test case LL/DDI/ADV/BV-22-C to remove previous Step 6 from test procedure and revise “repeat” instructions to reference the new step numbering.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11333 (rating 3): For test case LL/DDI/SCN/BV-19-C, removed the text before the Test Steps, modified step 9, added a row to Table 4.19, and modified the first Pass Verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11136 (rating 3): Updated Test Procedure steps 3 and 6 and Pass Verdict for test cases LL/TIM/SCN/BV-01-C – -03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11068 (rating 3): Replaced old MSC with new MSC for test cases LL/SEC/ADV/BV-02-C and -03-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11041 (rating 3): Updated Test Purpose, MSC, Test Procedure steps and Pass Verdict and added a Fail Verdict for test case LL/DDI/SCN/BV-37-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11312 (rating 3): Updated MSC and added step 8 to test procedure for test case LL/DDI/SCN/BV-16-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>TSE 11169 (rating 4): Combined test cases LL/PAC/SLA/BI-01-C and LL/PAC/MAS/BI-01-C under one section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TSE 10743 (rating 4): Updated test procedure steps and pass verdict items and added a fail verdict for test case LL/CON/MAS/BV-41-C and updated test procedure steps for test case LL/CON/SLA/BV-40-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11923 (rating 1): Updated test case LL/TIM/MAS/BV-07-C to change &quot;REQ&quot; to &quot;RSP&quot; in test procedure and MSC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11851 (rating 1): Added note after test procedure steps for test case LL/CON/SLA/BV-23-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11413 (rating 4): Added new section &quot;Extended Advertising, Accepting Connections with Random address,&quot; which includes new test cases LL/CON/ADV/BV-14-C – -16-C and updated TMCT accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11092 (rating 2): Updated Inconclusive Verdicts for test cases LL/CON/MAS/BV-53-C, LL/CON/SLA/BV-57-C (in lieu of LL/CON/MAS/BV-55-C due to a text change since the CR was drafted), and LL/CON/SLA/BV-55-C and -56-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11914 (rating 4): Added a pass verdict for the section containing test case LL/DDI/SCN/BV-19-C; for section previously containing only test case LL/DDI/SCN/BV-20-C, added new test cases</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>LL/DDI/SCN/BV-64-C</td>
<td>-66-C</td>
<td></td>
<td>added column to test configuration table, modified step 5 of the test procedure, and added pass verdicts; added pass verdict for test cases LL/DDI/SCN/BV-21-C, -23-C, -25-C, section containing -29-C (and -48-C -- -50-C), section containing -30-C (and -51-C -- -53-C), -35-C -- -37-C, section containing -38-C (and -60-C, -61-C), section containing LL/CON/SLA/BV-93-C -- -97-C and LL/CON/MAS/BV-89-C -- -93-C; modified section previously containing only LL/DDI/SCN/BV-24-C, renamed test case, and added new test case LL/DDI/SCN/BV-62-C and modified step 5 of the test procedure, added a pass verdict, and removed the inconclusive verdict; modified section previously containing only LL/DDI/SCN/BV-33-C, renamed test case, and added new test case LL/DDI/SCN/BV-63-C and added pass verdicts; modified section previously containing only LL/TIM/SCN/BV-01-C and -04-C, renamed test cases, and added new test cases LL/TIM/SCN/BV-05-C and -06-C and added pass verdicts; modified section previously containing only LL/TIM/SCN/BV-02-C, renamed test case, and added new test case LL/TIM/SCN/BV-07-C and added pass verdict; modified section previously containing only LL/TIM/SCN/BV-03-C, renamed test case, and added new test case LL/TIM/SCN/BV-08-C and added pass verdict. Updated TCMT accordingly.</td>
</tr>
<tr>
<td>LL/DDI/SCN/BV-29-C</td>
<td>-48-C -- -50-C</td>
<td></td>
<td>TSE 11761 (rating 3): For “AoD Connectionless CTE Scanning” section (now containing test cases LL/DDI/SCN/BV-29-C and -48-C -- -50-C) updated MSC with new from 11761 CR and applied update to accommodate TSE 11266 by removing the &quot;Repeat for LE 1M and LE 2M PHYs (where supported)&quot; text from the upper left of the Visio diagram, added step 15 to test procedure and updated step numbering in final step accordingly, and modified one pass verdict and added another pass verdict. For &quot;AoA Connectionless CTE Scanning&quot; section (now containing test cases LL/DDI/SCN/BV-30-C and -51-C -- -53-C) updated MSC with new from 11761 CR and applied update to accommodate TSE 11266 by removing the &quot;Repeat for LE 1M and LE 2M PHYs (where supported)&quot; text from the upper left of the Visio diagram, modified step 6 of the test procedure, added a new step after step 15 in the test procedure</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and updated number in final step accordingly, and modified one pass verdict and added another pass verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 9428 (rating 3): Updated test case LL/DDI/ADV/BV-26-C with a new MSC (modified to accommodate changes made in TSE 11266), revised test steps 2, 4, 16, and 17, and added a bullet to the Notes section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated test cases LL/DDI/ADV/BV-51-C – 54-C to change &quot;without ADI&quot; to &quot;ADI not allowed in scan response&quot; and &quot;with ADI&quot; to &quot;ADI allowed in scan response&quot; per TSE 11459 (new test cases added for TSE 11266). Updated test cases LL/DDI/ADV/BV-22-C and -50-C to swap CSA #2 and CSA #1 in test case name (in test procedures, TCMT entries, and TCRL entries).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Changes to test cases LL/CON/MAS/BV-72-C and -125-C and LL/CON/SLA/BV-76-C and -128-C to reference correct test cases to run, addressing issue created by an old typo that was carried into newly created test cases for TSE 11266.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12142 (rating 2): Added missing test cases LL/DDI/ADV/BI-05-C and -06-C and</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LL/DDI/SCN/BV-49-C to TCMT and deleted test case LL/TIM/SCN/BV-45-C from TCMT.</td>
</tr>
<tr>
<td>16</td>
<td>5.1.1</td>
<td>2019-08-01</td>
<td>Approved by BTI. Prepared for TCRL 2019 publication.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12216 (CR file in comment 49132). Modified “Power Change Indication on PHY Change” section by replacing MSC and all test steps and moving test cases into a table, also adding new test cases LL/PCL/SLA/BV-37-C – -39-C and LL/PCL/MAS/BV-37-C – -39-C and replacing Pass Verdict text. Modified “Power Control Request when a CIS has been established” section by replacing MSC and updating and adding test steps; updated Pass Verdict with new step number. Modified “Power Control Response when a CIS has been established” section by replacing MSC and updating and adding test steps; updated test step numbers in Pass Verdict. Modified “Remote Transmit Power Level Request” section with new text in step 3 and added Fail Verdict. Updated TCMT accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12336 (CR file in comment 49550). Modified “Power Configuration Response, Min and Max Power Level Reached” section by renaming section, modifying Test Purpose, replacing the MSC and all test steps, and replacing the Pass Verdict. Added new section “Max and Min Power Level Response at Max and Min Power”, including test cases LL/PCL/SLA/BV-40-C and LL/PCL/MAS/BV-40-C. Updated TCMT accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12349 (CR in comment 48781). Modified “Path Loss Monitoring” section by replacing MSC, adding test steps 9–11, and updating Pass Verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12341 (CR in “Comments Addressed” file in comment 49240). Revised wording in BIS/BRD section, updated wording in Test Purpose, test steps, test case table, and Pass Verdict for “Broadcast Isochronous Stream Setup”</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>section. Revised wording in BIS/SNC section; updated wording in test steps and test case table in “Broadcast Isochronous Stream Synchronization Setup” section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12425 (CR in “Proposed Changes” file in comment 49743). Replaced MSCs for test cases LL/CIS/MAS/BV-10-C and -11-C. Modified text in test step 6 in “Sending data in Unidirectional CIS” section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12465 (CR in “Proposed change” file in Supporting Files section). Updated tables in “Common PDU Contents” section, adding the RFU field and moving BaseCRCInit to the next row down.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12466 (CR in “Proposed change” file in Supporting Files section). Added text to beginning of CIS “Timing Requirements” section. Corrected typo in list item 1 of same section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12468 (CR file in comment 49871). Added new section “Sending and Receiving Unframed Empty PDUs with LLID=0b01, CIS” including new test cases LL/IAL/MAS/BV-45-C and LL/IAL/SLA/BV-45-C, new section &quot;LL/IAL/BRD/BV-TBD52-C [Sending Unframed Empty PDUs with LLID=0b01, BIS]&quot; and new section &quot;LL/IAL/SNC/BV-29-C [Receiving Unframed Empty PDUs with LLID=0b01, BIS]&quot; and updated TCMT accordingly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12591 (CR file in Supporting Files section). Updated test procedure for new “Power Control Request when a CIS has been established” section to move previous steps 2 and 3 to be steps 1 and 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11954 (rating 1): Changed a reference in the Test Realization section to refer to the correct HCI volume and part.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12032 (rating 1): Replaced MSCs for test cases LL/SEC/ADV/BV-03-C, -16-C, and -19-C to address an incorrect command name in the figures.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12334 (rating 1): Changed “EventCounter” to “PeriodicEventCounter” in “AoD Connectionless CTE Scanning” and “AoA Connectionless CTE Scanning” test steps to align with a change made in erratum 12325.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12108 (rating 2): Updated the “Optional Test Steps” section under “Common Test Procedure Steps” to include the LL Minimum Number of Used Channels procedure as an optional test step.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11172 (rating 3): Updated test purpose, MSC, and pass verdict, and added test steps 7–11 for test case LL/DDI/SCN/BV-13-C; updated test purpose, MSC, test step 5, and pass verdict, and added test steps 11–17 for test case LL/SEC/ADV/BV-05-C; updated test purpose, MSC, and pass verdict, and added test steps 9–14 for test case LL/SEC/ADV/BV-08-C; updated test purpose, MSC, and pass verdict, and added test steps 9–14 for test case LL/SEC/ADV/BV-11-C to align with erratum 10684, whereby Link Layer only resolves an RPA when address resolution is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11818 (rating 3): Updated test procedures for “AoD Connectionless CTE Scanning, Incorrect CRC” (affecting test cases LL/DDI/SCN/BV-31-C and -54-C – -56-C) and “AoA Connectionless CTE Scanning, Incorrect CRC” (affecting test cases LL/DDI/SCN/BV-32-C and -57-C – -59-C) sections to address issue with generating a periodic advertising reports.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12294 (rating 3): Updated test case LL/DDI/SCN/BV-35-C to fix incorrect repeat of a step.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12110 (rating 1): Fixed references to align with changes made in erratum 11876.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per Issue 12337 (CR from comment 49925): Updated MSCs, test steps, and pass verdicts for test cases LL/BIS/BRD/BV-01-C, -02-C, -08-C, and -10-C.</td>
</tr>
</tbody>
</table>
|                    |                  |      | Updated per issue 12519 (CR from comment 50010): Updated MSC for “Test Command
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Generated Isochronous SDUs Optional Test Steps in the Optional Test Steps section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per issue 12404 (CR from comment 50075): For section containing test cases LL/CIS/MAS/BV-01-C, -02-C, -25-C, -31-C, and -32-C, updated MSC (part B only); revised test procedure by adding a new step 8 and revising the final two steps; updated the pass verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12662 (rating 2): Moved test case LL/DDI/SCN/BV-66-C to a different location in the TCMT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12497 (rating 4): Updated test step 10 for test case LL/CON/MAS/BV-73-C; updated test step 1 and rounds table for test case LL/CON/MAS/BV-74-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per issue 12456 (CR in comment 51127): Updated test purpose, MSC, test steps, and pass verdict for “Remote Transmit Power Level Request” section, affecting test case LL/PCL/SLA/BV-29-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per issue 12612 (CR in comment 51253): Replaced the MSCs for the “Power Control Response” section (affecting test cases LL/PCL/SLA/BV-08-C – -11-C and LL/PCL/MAS/BV-08-C – -11-C), the “Power Control Response when a CIS has been established” section (affecting test cases LL/PCL/SLA/BV-25-C and -28-C and LL/PCL/MAS/BV-25-C and -27-C), and the “Power Control Response with RF Path Compensation” section (affecting test cases LL/PCL/SLA/BV-33-C – -36-C and LL/PCL/MAS/BV-33-C – -36-C).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated per issue 12689 (CR in comment 51284): Updated MSC and test steps 2, 3, 6, 8, 10, and 12 in the “Power Control Response when a CIS has been established” section (affecting test cases LL/PCL/SLA/BV-25-C and -28-C and LL/PCL/MAS/BV-25-C and -27-C).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11245 (rating 3): Updated test step 5 and expected outcome section for “Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)” section, affecting test cases LL/DDI/SCN/BV-24-C and -62-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 11534 (rating 3): Updated references, test steps, and the MSC for test case LL/DDI/SCN/BV-15-C.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>TSE 11429 (rating 3):</td>
<td>Updated “Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address” section to include test cases LL/CON/INI/BV-18-C (preexisting) and -24-C (moved from an independent section). Updated MSC and put both TCs in a TC table. Updated TCMT accordingly.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAL restructure: Removed “Isochronous Adaptation Layer” item from Test Suite Structure diagram in Test Strategy section. Removed Isochronous Streams paragraph from Test Strategy section. Removed Isochronous Adaptation Layer paragraph from Test Groups section. Removed IAL from Test Case Naming Conventions abbreviation table. Removed entire IAL section (all related test cases). Removed all IAL items from the TCMT.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue 12639 (CR in comment 51609):</td>
<td>For “Power Control Response” section (test cases LL/PCL/SLA/BV-08-C – -11-C and LL/PCL/MAS/BV-08-C – -11-C), updated initial condition, MSC, all steps of the test procedure, and the expected outcome; added Example test procedure to Notes section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue 12741 (CR in comment 51811):</td>
<td>For “Power Control Request – Initiate” section (test cases LL/PCL/SLA/BV-03-C – -05-C and LL/PCL/MAS/BV-03-C – -05-C), updated test steps 1 and 4 and updated test case table to modify entries for test cases LL/PCL/SLA/BV-05-C and MAS/BV-05-C and to delete items LL/PCL/SLA-06-C and MAS/BV-06-C (updated TCMT accordingly). For “Power Control Response” section (test cases LL/PCL/SLA/BV-08-C – -11-C and LL/PCL/MAS/BV-08-C – -11-C), added new item to Notes section. For &quot;Power Control Request when a CIS has been established&quot; section (test cases LL/PCL/SLA/BV-20-C and -22-C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and LL/PCL/MAS/BV-20-C and -23-C, updated test step 4 and added test step 10, added a note to the Notes section, and updated test case table to modify entries for test cases LL/PCL/SLA/BV-22-C and MAS/BV-23-C. For “Power Control Response when a CIS has been established” section (test cases LL/PCL/SLA/BV-25-C and -28-C and LL/PCL/MAS/BV-25-C and -27-C). added a note to the Notes section. For “Power Control Response with RF Path Compensation” section (test cases LL/PCL/SLA/BV-33-C – -36-C and LL/PCL/MAS/BV-33-C – -36-C), added a note to the Notes section. TSE 11661 (rating 1): Updated initial condition with a clarifying note for test case LL/TIM/MAS/BV-07-C. TSE 11629 (rating 4): Added a new “Constant Tone Extension Disabled on PHY Change” section featuring new test cases LL/CON/MAS/BV-131-C and SLA/BV-134-C. Updated TCMT accordingly. TSE 11177 (rating 4): Added a new section for test case LL/DDI/ADV/BV-61-C. Updated TCMT accordingly. TSE 12653 (rating 3): For “Data Length Update – Peer Does Not Support LE Coded PHY” section, updated MSC and test steps 1, 12, and 14; simplified and renamed test case table and deleted test cases LL/CON/SLA/133-C and MAS/BV-130-C (updated TCMT accordingly); updated pass verdict and removed fail verdict. TSE 12562 (rating 3): Updated MSC and test step 2 and added a new step 3 for section test case LL/CON/INI/BV-23-C. TSE 12114 (rating 2): In Optional Test Steps section, replaced MSC and updated the description/instructions for “LE PHY Update Procedure (LE Coded Switch) Optional Test Steps” section. Issue 12308 (CR from comment 51275): For test case LL/CIS/MAS/BV-30-C, updated MSC from part A/part B setup to a single combined MSC; removed test steps; added a pass verdict. Issue 12355 (CR from comment 51017): For “CIS Setup Procedure, Master Initiated” section, updated the MSC and a test step, and added a pass verdict. For the “Connected Isochronous Stream Using Non-</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test Command, Master Initiated” section, updated a test procedure and pass verdict and replaced the part C MSC. For the “CIS Setup Response Procedure, Slave” section, updated a test procedure, the MSC, and the pass verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12431 (CR from comment 50784): For CIS_Offset_Min and CIS_Offset_Max issue: deleted text from “State: Connected Isochronous Stream, Slave” section; updated test steps and pass verdict for the “CIS Setup Procedure, Master Initiated” and “Connected Isochronous Stream Using Non-Test Command, Master Initiated” sections (also MSC for latter); deleted text and removed associated table rows from several sections in the CIS section that dealt with the CIS_Offset_Min/Max issue.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12488 (CR from comment 51611): For “CIS Setup Procedure, Master Initiated” section, replaced MSC, removed and renumbered test steps, and removed and renumbered pass verdicts; for “Connected Isochronous Stream Using Non-Test Command, Master Initiated” section, collapsed MSC into single figure and updated and renumbered test steps and pass verdicts; for test case LL/CIS/SLA/BV-02-C, replaced MSC, updated test steps, updated pass verdicts; in “Isochronous Testing” section, completely removed sections for test cases LL/IST/ERR/BI-01-C and -02-C (updated TCMT and TCRL accordingly).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12489 (CR from comment 51621): For “Broadcast Isochronous Stream Using Non-Test Command” section, replaced MSCs and updated test steps and pass verdicts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12520 (CR from comment 51147): For section “Sending data in Unidirectional CIS”, revised the MSC and updated the test steps and pass verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12567 (CR from comment 51771): For “Sending and Receiving Data in Bidirectional CIS” section, revised initial condition, state variable tables, test steps, and pass verdict and replaced the MSC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12593 (CR from comment 51274): In “Default Values for Set CIG Parameters Commands” section, updated Note 1 at the end of the parameters table.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12597 (CR from comment 51219): Updated state variable table for test case LL/CIS/MAS/BV-09-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12514 (CR from comment 51314): Updated TCMT to include an internal heading for “Isochronous Streams Procedures”.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12674 (CR from comment 51270): For test case LL/BIS/BRD/BV-08-C, added an initial condition, updated the test steps, and added an inconclusive verdict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12697 (CR from comment 51982): Added a new ISOC item to the Common Test Procedure Steps, Optional Test Steps section; for test cases LL/BIS/BRD/BV-04-C, -09-C, -10-C and -16-C, replaced MSC and updated test steps and pass verdict; for test case LL/BIS/BRD/BV-05-C and -10-C, replaced MSC and updated test steps; for “Broadcast Isochronous Stream Using Non-Test Command” section, replaced MSC part B and revised test steps and pass verdict; for test case LL/BIS/SNC/BV-09-C, replaced MSC and updated test step (updated test procedure, test steps, and pass verdict to match CR, as text was not aligning).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 12452 (rating 2): Updated test step and added an inconclusive verdict for section containing test cases LL/PAC/MAS/BI-01-C and LL/PAC/SLA/BI-01-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue 12704 (CR from comment 51911): For section “Broadcast Isochronous Stream Synchronization Setup”, updated initial condition, test steps, and pass verdict, and replaced MSC.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
</tbody>
</table>
|                    |                  |      | Issue 12708 (CR from comment 51831): Added a pass verdict to sections “Connected Isochronous Stream Using Non-Test Command, Master Initiated” and “Broadcast Isochronous Stream Using Non-Test Command”.
|                    |                  |      | Issue 12717 (CR from comment 51216): For test case LL/CIS/SLA/BV-22-C, modified a test step and added a pass verdict.
|                    |                  |      | Issue 12767 (CR from comment 51924): Replaced MSCs for sections “Acknowledgement Scheme, Master” and “Acknowledgement Scheme, Slave”.
|                    |                  |      | TSE 11029 (rating 3): Updated MSC and test steps for test cases LL/CON/INI/BV-14-C and -15-C.
|                    |                  |      | Fixed Milan references to V 5.2. Integration review feedback incorporated to fix issues related to original CR for 11818.
|                    |                  |      | TSE 12924: Globally fixed “Lower/Upper Tester expects” types of wording to “Lower/Upper Tester receives” types of wording where appropriate.
|                    |                  |      | Integration review feedback resolving .X and Milan references to real numbers and making another fix related to 11818.
|                    |                  |      | Incorporated review feedback on TSE 11818 following TSE CRv6.
|                    |                  |      | TSE 12930 (rating 2): Removed “OR LL 11/1” from the TCMT for test case LL/DDI/SCN/BV-47-C.
|                    |                  |      | Revised document numbering convention, setting last release publication of 5.1.1 as p16; added publication number column to Revision History.
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>p17</td>
<td>2020-01-07</td>
<td>Updated Contributors list and fixed duplicates and alphabetization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Integration review feedback to fix TSPX notations for some IXIT items not previously defined.</td>
</tr>
</tbody>
</table>

**Contributors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antonio Narváez Castillo</td>
<td>AT4 wireless</td>
</tr>
<tr>
<td>Juan Manuel Hidalgo</td>
<td>AT4 wireless</td>
</tr>
<tr>
<td>Francisco Narvaez</td>
<td>AT4 wireless</td>
</tr>
<tr>
<td>Elisa Rincón</td>
<td>AT4 wireless</td>
</tr>
<tr>
<td>Angel Romero</td>
<td>AT4 wireless</td>
</tr>
<tr>
<td>Norbert Grunert</td>
<td>Broadcom</td>
</tr>
<tr>
<td>Angel Polo</td>
<td>Broadcom</td>
</tr>
<tr>
<td>Bogdan Alexandru</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Alexandru Andreescu</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Nathan Burns</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Gene Chang</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Jeff Drake</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Tharon Hall</td>
<td>Cloud2GND LLC</td>
</tr>
<tr>
<td>Mayank Batra</td>
<td>CSR</td>
</tr>
<tr>
<td>Joe Decuir</td>
<td>CSR</td>
</tr>
<tr>
<td>Giriraj Goyal</td>
<td>CSR</td>
</tr>
<tr>
<td>Robin Heydon</td>
<td>CSR</td>
</tr>
<tr>
<td>Neil Stewart</td>
<td>CSR</td>
</tr>
<tr>
<td>Name</td>
<td>Company</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Fabien Duvoux</td>
<td>Ellisys</td>
</tr>
<tr>
<td>Kyle Penri-Williams</td>
<td>Ellisys</td>
</tr>
<tr>
<td>Clement Vacheron</td>
<td>Ellisys</td>
</tr>
<tr>
<td>Harish Balasubramaniam</td>
<td>Intel</td>
</tr>
<tr>
<td>Magnus Eriksson</td>
<td>Intel</td>
</tr>
<tr>
<td>Oren Haggai</td>
<td>Intel</td>
</tr>
<tr>
<td>Yao Wang</td>
<td>IVT Corporation</td>
</tr>
<tr>
<td>Rajan S Pallathu</td>
<td>MindTree Ltd.</td>
</tr>
<tr>
<td>Mika Kasslin</td>
<td>Nokia</td>
</tr>
<tr>
<td>Jukka Reunamäki</td>
<td>Nokia</td>
</tr>
<tr>
<td>David Engelien-Lopes</td>
<td>Nordic Semiconductor ASA</td>
</tr>
<tr>
<td>Brian A. Redding</td>
<td>Qualcomm</td>
</tr>
<tr>
<td>Joel Linsky</td>
<td>Qualcomm Atheros</td>
</tr>
<tr>
<td>Magnus Sommannsson</td>
<td>Qualcomm Technologies International, Ltd.</td>
</tr>
<tr>
<td>Jean-Philippe Lambert</td>
<td>RivieraWaves</td>
</tr>
<tr>
<td>Rasmus Abildgren</td>
<td>Samsung Electronics</td>
</tr>
<tr>
<td>Rogier Scheffer</td>
<td>STMicroelectronics</td>
</tr>
<tr>
<td>Paul Vanoostende</td>
<td>STMicroelectronics</td>
</tr>
<tr>
<td>Ben Brown</td>
<td>Teledyne LeCroy</td>
</tr>
<tr>
<td>Tony Viscardi</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>Martti Söderlund</td>
<td>TietoEnator</td>
</tr>
<tr>
<td>Jason Hillyard</td>
<td>Wicentric</td>
</tr>
</tbody>
</table>