Audio/Video Control Transport Protocol (AVCTP)

Abstract:
This document specifies the Conformance Tests of Audio/Video Control Transport Protocol (AVCTP).
### Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>2003-02-01</td>
<td>Release for Voting Draft</td>
</tr>
<tr>
<td>Version 1.0</td>
<td>2003-05-01</td>
<td>Updated title and header</td>
</tr>
<tr>
<td>Version 1.2</td>
<td>2004-04-14</td>
<td>Updated Disclaimer and Copyright Notice. Clerical changes.</td>
</tr>
<tr>
<td>1.1.1r1</td>
<td>2004-12-29</td>
<td>Editorial and format change.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 700 for TP/CCM/BI-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 702 for TCMT.</td>
</tr>
<tr>
<td>1.2.1</td>
<td>2005-02-10</td>
<td>Document number change.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 703 for TP/NFR/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 708 for TCMT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 709 for TCMT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 710 for TP/FRA/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 711 for TP/FRA/BV-04-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorporate TSE 712 for TP/NFR/BI-01-C</td>
</tr>
<tr>
<td>1.2.2r1-02 1.2.2r2</td>
<td>2005-08-15–2005-11-03</td>
<td>Incorporate TSE 783 for TP/FRA/BV-01-C, TP/FRA/BV-03-C, TP/FRA/BV-04-C</td>
</tr>
<tr>
<td>1.2.2</td>
<td></td>
<td>Correct TSE 741 for TP/NFR/BV-01-C and TP/NFR/BV-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corrections to TP/CCM/BI-01-C in response to AV WG input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corrections to Test Procedure for TP/NFR/BV-03-C, Corrected cross refs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for TP/NFR/BV-04-C/TP/FRA/BV-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prepare for publication</td>
</tr>
<tr>
<td>1.2.3r0</td>
<td>2006-11-01</td>
<td>Add Conformance section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSE 1814: TP/CCM/BV-03-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSE 1865: Fix graphic for TP/CCM/BV-04-C</td>
</tr>
<tr>
<td>Revision Number</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.2.3r1</td>
<td>2006-11-30</td>
<td>Accept changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input reviewer's comments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Remove table of abbreviations, list of tables, and list of figures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Move Section 5, References, to Section 2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Move MSCs to their respective test cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Merge table cells for selection criteria that pertain to a single test case</td>
</tr>
<tr>
<td>1.2.3</td>
<td>2006-01-08</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.2.4r0</td>
<td>2008-02-01</td>
<td>TSE 2391: TP/FRA/BV-01-C, TP/FRA/BV-02-C</td>
</tr>
<tr>
<td>1.2.4</td>
<td>2008-04-01</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.2.5r0</td>
<td>2008-10-01</td>
<td>TSE 2593 TP/NFR/BI-01-C, MSC correction</td>
</tr>
<tr>
<td>1.2.5</td>
<td>2008-11-25</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.3.6r0-r2</td>
<td>2010-12-17 - 2011-03-01</td>
<td>TSE 3820 TP/CCM/BV-01-C, TP/CCM/BV-02-C, TP/CCM/BV-03-C, TP/CCM/BV-04-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Accepted previous changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changed version number in previous row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input Alicia C's comments for TP/CCM/BV-01-C, TP/CCM/BV-02-C per TSE 3820</td>
</tr>
<tr>
<td>1.3.6</td>
<td>2011-07-21</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.3.7r0</td>
<td>2011-09-13</td>
<td>TSE 4409: TCMT for TP/NFR/BV-01-C, TP/NFR/BV-02-C</td>
</tr>
<tr>
<td>1.3.7r1</td>
<td>2011-10-25</td>
<td>Correction to TP/NFR/BV-01-C in TCMT by AC</td>
</tr>
<tr>
<td>1.3.7</td>
<td>2012-03-30</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.3.8r0</td>
<td>2012-05-18</td>
<td>TSE 4607: TP/NFR/BI-01-C: update MSC to remove “Data” from response.</td>
</tr>
<tr>
<td>1.4.0r0</td>
<td>2012-05-31</td>
<td>Updated specification revision labeling. Application of BTI recommendations for test case bulleting, indenting and fail verdict text. Corrected hypertext font coloring.</td>
</tr>
</tbody>
</table>
### Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.0r1</td>
<td>2012-06-06</td>
<td>Updated Section 2 to comply with latest Test Specification template.</td>
</tr>
<tr>
<td>1.4.0r2</td>
<td>2012-06-19</td>
<td>Edited all test cases to comply with latest test case template. Updated some MSCs that had missing function parameters or mismatching callback function names. Added relevant pass verdicts to some test cases. Split TCMT into two tables: one for combined CT/TG support and the other for individual CT or TG support. Remapped test cases to reference ICS items 2/2 to 2/7 which were previously unreferenced in the TCMT.</td>
</tr>
<tr>
<td>1.4.0</td>
<td>2012-07-24</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>1.4.1r01</td>
<td>2017-01-27</td>
<td>Converted test specification template.</td>
</tr>
<tr>
<td>1.4.1r02</td>
<td>2017-04-05</td>
<td>Converted to new Test Case ID conventions as defined in TSTO v4.1</td>
</tr>
<tr>
<td>1.4.1</td>
<td>2017-07-03</td>
<td>Approved by BTI. Prepared for TCRL 2017-1 publication.</td>
</tr>
</tbody>
</table>

### Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Trainor</td>
<td>CSR</td>
</tr>
<tr>
<td>Fisseha Mekuria</td>
<td>Ericsson</td>
</tr>
<tr>
<td>Morgan Lindqvist</td>
<td>Ericsson</td>
</tr>
<tr>
<td>Michinori Masuda</td>
<td>Matsushita Electric Industrial</td>
</tr>
<tr>
<td>Tsuyoshi Okada</td>
<td>Matsushita Electric Industrial</td>
</tr>
<tr>
<td>Christian Bouffioux</td>
<td>Philips</td>
</tr>
<tr>
<td>Henk Koopmans</td>
<td>Philips</td>
</tr>
<tr>
<td>Emmanuel Mellery</td>
<td>Philips</td>
</tr>
<tr>
<td>Piotr Polak</td>
<td>Philips</td>
</tr>
<tr>
<td>Harumi Kawamura</td>
<td>Sony</td>
</tr>
<tr>
<td>Hiroyasu Nouchi</td>
<td>Sony</td>
</tr>
<tr>
<td>Rudiger Mosig</td>
<td>Sony</td>
</tr>
<tr>
<td>Yoshinari Kumaki</td>
<td>Toshiba</td>
</tr>
<tr>
<td>Ichiro Tomoda</td>
<td>Toshiba</td>
</tr>
</tbody>
</table>
Contents

1 Scope ........................................................................................................................................ 8

2 References, Definitions, and Abbreviations ........................................................................... 9
  2.1 References ............................................................................................................................ 9
  2.2 Definitions ........................................................................................................................... 9
  2.3 Abbreviations ...................................................................................................................... 9

3 Test Suite Structure (TSS) ..................................................................................................... 10
  3.1 Test Strategy ....................................................................................................................... 10
  3.1.1 Introduction .................................................................................................................... 10
  3.1.2 Test Environment .......................................................................................................... 10
  3.1.3 Test Specifications Guidelines ...................................................................................... 10
  3.2 Test Suite Structure ........................................................................................................... 10
  3.3 Test Groups ....................................................................................................................... 11

4 Test Cases (TC) ..................................................................................................................... 12
  4.1 Introduction ......................................................................................................................... 12
  4.1.1 Test Case Identification Conventions ........................................................................... 12
  4.1.2 Conformance .................................................................................................................. 12
  4.1.3 Pass/Fail Verdict Conventions ...................................................................................... 13
  4.1.4 Assumptions .................................................................................................................. 13
  4.2 Connection Channel Management .................................................................................... 13
  4.2.1 Connection Channel Establishment request ............................................................... 13
    AVCTP/CT/CCM/BV-01-C ...................................................................................................... 13
    AVCTP/TG/CCM/BV-01-C ...................................................................................................... 13
  4.2.2 Disconnect a Channel ..................................................................................................... 16
    AVCTP/CT/CCM/BV-02-C ...................................................................................................... 16
    AVCTP/TG/CCM/BV-02-C ...................................................................................................... 16
  4.2.3 Connection Channel Establishment Response .............................................................. 18
    AVCTP/CT/CCM/BV-03-C ...................................................................................................... 18
    AVCTP/TG/CCM/BV-03-C ...................................................................................................... 18
  4.2.4 Disconnect Response ...................................................................................................... 20
    AVCTP/CT/CCM/BV-04-C ...................................................................................................... 20
    AVCTP/TG/CCM/BV-04-C ...................................................................................................... 20
  4.2.5 AVCTP/CT/CCM/BV-01-C [Connection Channel Establishment - AVCTP Connection Already Present] .................................................................................................................. 22
  4.3 Non-Fragmented Messages .............................................................................................. 24
  4.3.1 AVCTP/CT/NFR/BV-01-C [CT Sends Non-Fragmented Command Message] .............. 24
  4.3.2 AVCTP/TG/NFR/BV-02-C [TG Sends Non-Fragmented AVCTP Response Message] .... 26
  4.3.3 AVCTP/TG/NFR/BV-03-C [TG Receives Incoming command] .................................. 28
  4.3.4 AVCTP/CT/NFR/BV-04-C [CT Sends Non-Fragmented Command] ............................ 30
  4.3.5 AVCTP/TG/NFR/BV-01-C [TG Handles Control Message with Invalid PID] ............... 33
  4.4 Fragmented Messages ....................................................................................................... 34
  4.4.1 AVCTP/CT/FRA/BV-01-C [CT Fragments Command Messages] .............................. 34
  4.4.2 AVCTP/TG/FRA/BV-02-C [TG Fragments Response Messages] ............................... 36
  4.4.3 AVCTP/TG/FRA/BV-03-C [TG Reassembles Fragmented Command Message] .......... 39
  4.4.4 AVCTP/CT/FRA/BV-04-C [CT Reassembles a Fragmented Response Message] ....... 41

5 Test Case Mapping ................................................................................................................. 45
1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and Test Cases (TC) to test the Audio/Video Control Transport Protocol (AVCTP).

The objective of this document is to provide a basis for the conformance tests 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.


[5] Bluetooth Core Specification v2.0 or later


2.2 Definitions
For the purpose of this Bluetooth document, the definitions from [3], [5], and [6] apply.

2.3 Abbreviations
For the purpose of this Bluetooth document, the abbreviations from [3], [5], and [6] apply.
3 Test Suite Structure (TSS)

3.1 Test Strategy

3.1.1 Introduction

The tests are organized through a formal test suite.

For the conformance tests, the qualification process is performed against a Lower Tester whose requirements are derived from the test suite specifications expressed in the present document.

3.1.2 Test Environment

Figure 3.1 provides a schematic view of the test environment required for executing the AVCTP test suite.

![AVCTP Test Model](image)

*Figure 3.1: AVCTP Test Model*

Two entities are defined around the IUT: The Upper Tester and the Lower Tester.

Both are referred to in the message sequence charts (MSC) describing the test procedures in their respective test cases in Sections 4.2, 4.3, and 4.4.

3.1.3 Test Specifications Guidelines

The following guidelines drive the specifications of the test suite provided in this document:

The tests are defined in such a way that their execution and the interpretation of the test results can be automated to the maximum extent.

The tests cover both valid and invalid behavior.

3.2 Test Suite Structure

This section defines the tree structure of the conformance tests specified for AVCTP.
The test suite structure (TSS) is presented in Figure 3.2.

The TSS is composed of nested test groups organized in a top down approach.

**AVCTP TSS**

- Connection channel management
- Non-fragmented messages
- Fragmented messages

*Figure 3.2: AVCTP Test Suite Structure*

### 3.3 Test Groups

The test groups are organized in 2 levels.

The first level is dedicated to the type of message conveyed (fragmented or non-fragmented) and to the management of a connection channel.

The second (last) level in each branch contains the standard ISO groups BV and BI.
4 Test Cases (TC)

4.1 Introduction

4.1.1 Test Case Identification Conventions

Test cases shall be assigned unique identifiers per the conventions in [6]. The convention used here is 
<spec abbreviation>/<IUT role>/<class>/<feat>/<func>/<subfunc>/<cap>/<xx>-<nn>-<yy>.

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

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Spec Identifier &lt;spec abbreviation&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVCTP</td>
<td>Audio/Video Control Transport</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Role Identifier &lt;IUT role&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>TG</td>
<td>Target role</td>
</tr>
<tr>
<td>CT</td>
<td>Controller role</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Feature Identifier &lt;feat&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM</td>
<td>Connection channel management</td>
</tr>
<tr>
<td>FRA</td>
<td>Fragmented message</td>
</tr>
<tr>
<td>NFR</td>
<td>Non-fragmented message</td>
</tr>
</tbody>
</table>

Table 4.1: AVCTP TC Feature Naming Conventions

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 Specification, 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.

4.1.3 Pass/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.

The convention in this test specification is that, unless there are a specific set of fail conditions outlined in the test case, the IUT fails the test case as soon as one of the pass criteria conditions cannot be met. If this occurs the outcome of the test shall be the Fail Verdict.

4.1.4 Assumptions

Only one point-to-point configuration is considered.

To execute the AVCTP test suite, it is required that the Bluetooth reference stack layers below AVCTP on the IUT are in conformance with their respective specifications.

The IUT can play both the role of CT and TG. The selected role is specified for each test case.

An ACL link has been set-up between the IUT and the test system.

4.2 Connection Channel Management

Test Group Objective:

• To verify the correct implementation of the service primitives managing the connection channel on both CT and TG sides.

4.2.1 Connection Channel Establishment request

• Test Purpose

To verify that the IUT (CT and TG) is able to request a connection channel establishment. Note that testing of L2CAP is out of scope.

• Test Case ID(s)

AVCTP/CT/CCM/BV-01-C
AVCTP/TG/CCM/BV-01-C

• Reference

[3] 11.2.1
• Initial Condition
  - No L2CAP channel with the AVCTP PSM has yet been set-up between the IUT and the Lower Tester.
  - No limitations apply to the configuration of the L2CAP channel.
  - The PID identifying the profile using AVCTP is defined as PID\textsubscript{Test\_System}.
  - No profile with PID = PID\textsubscript{Test\_System} is yet registered in the IUT.

The BD_ADDR of the Lower Tester is defined as BD_ADDR\textsubscript{Lower\_Tester}.

• Test Procedure
  1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration, the Upper Tester registers the function ConnectCfm\_CB\textsubscript{Test\_System} for callback on the AVCT\_Connect\_Cfm event by sending an AVCT\_EventRegistration command to the IUT with the following input parameter values:
     \begin{align*}
     \text{Event} & = \text{AVCT\_Connect\_Cfm} \\
     \text{Callback} & = \text{ConnectCfm\_CB}\textsubscript{Test\_System} \\
     \text{PID} & = \text{PID}\textsubscript{Test\_System}
     \end{align*}
     After sending this command the Upper Tester waits on the response from the IUT.
  2. If the Upper Tester and IUT conform to the AVCTP upper interface for connection, the Upper Tester sends an AVCT\_ConnectReq command to the IUT with the following input parameter values:
     \begin{align*}
     \text{BD\_ADDR} & = \text{BD\_ADDR}\textsubscript{Lower\_Tester} \\
     \text{PID} & = \text{PID}\textsubscript{Test\_System}
     \end{align*}
     If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for connection, the profile using the IUT sends an appropriate L2CAP\_ConnectReq to the Lower Tester.
  3. Upon receipt of an L2CAP\_ConnectReq from the IUT the Lower Tester sends an appropriate L2CAP\_ConnectRsp to the IUT.
  4. Upon receipt of an L2CAP\_ConfigReq from the IUT the Lower Tester sends an appropriate L2CAP\_ConfigRsp to the IUT.
  5. The Lower Tester sends an L2CAP\_ConfigReq to the IUT.
One ACL link exists between the IUT and the Test System. 
No L2CAP channel exists between the IUT and the Test System.

**Figure 4.1: AVCTP/CT/CCM/BV-01-C & AVCTP/TG/CCM/BV-01-C**

- **Expected Outcome**

  **Pass verdict:**

  If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration:

  The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

  Result = 0x0000 (Event successfully registered)
The IUT calls the ConnectCfm_CB function in the Upper Tester with the following parameters:

- **BD_ADDR** = BD_ADDR_Lower_Tester
- **Connect Result** = 0x0000 (L2CAP Connect Request successful)
- **Config Result** = 0x0000 (L2CAP Configure successful)
- **Status** = L2CAP Connect Request Status

If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection, the IUT returns the following AVCT_ConnectReq output parameter values to the Upper Tester:

RSP = 0x0000 (Request accepted)

The IUT sends an L2CAP_ConnectReq to the Lower Tester with the PSM field containing the PSM value assigned to AVCTP.

Upon receipt of the L2CAP_ConnectRsp from the Lower Tester, the IUT sends an L2CAP_ConfigReq to the Lower Tester with the parameter values specified in the AVCT_ConnectReq function call or supplied by the profile using the IUT.

Upon receipt of the L2CAP_ConfigReq from the Lower Tester, the IUT sends the L2CAP_ConfigRsp to the Lower Tester with the specified parameter values (testing of L2CAP is out of scope).

### 4.2.2 Disconnect a Channel

- **Test Purpose**

  To verify that the IUT (CT and TG) is able to disconnect a channel previously established.

- **Test Case ID(s)**

  - AVCTP/CT/CCM/BV-02-C
  - AVCTP/TG/CCM/BV-02-C

- **Reference**

  [3] 11.2.3

- **Initial Condition**

  - An L2CAP channel with the AVCTP PSM has been set-up between the IUT and the test system.
  - The PID identifying the profile using AVCTP is defined as PIDTest_System.
  - Only one profile with PID = PIDTest_System is registered in the IUT (e.g. Error! Reference source not found. or Error! Reference source not found. has been successfully executed).
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR_Lower_Tester.

- **Test Procedure**

  1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection event registration, the Upper Tester registers the function DisconnectCfm_CB function for callback on
the AVCT_Disconnect_Cfm event by sending an AVCT_EventRegistration command to the IUT with the following parameter values:

- **Event** = AVCT_Disconnect_Cfm
- **Callback** = DisconnectCfm_CB\_{Test\_System}
- **PID** = PID_{Test\_System}

After sending this command the Upper Tester waits on the response from the IUT.

2. If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection, the Upper Tester sends an AVCT_DisconnectReq command to the IUT with the following parameter values:

   - **BD_ADDR** = BD ADDR\_{Lower\_Tester}
   - **PID** = PID_{Test\_System}

If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for disconnection the profile using the IUT sends an appropriate L2CAP_DisconnectReq to the Lower Tester.

3. Upon receipt of an L2CAP_DisconnectReq from the IUT the Lower Tester sends an appropriate L2CAP_DisconnectRsp to the IUT.

---

**Figure 4.2: AVCTP/CT/CCM/BV-02-C & AVCTP/TG/CCM/BV-02-C**

- **Expected Outcome**
  - **Pass verdict:**
If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection event registration:

The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

Result = 0x0000 (Event successfully registered)

The IUT calls the DisconnectCfm_CB function in the Upper Tester with the following parameter values:

\[
\begin{align*}
\text{BD_ADDR} & = \text{BD_ADDR}_{\text{Lower Tester}} \\
\text{Disconnect Result} & = 0x0000 \text{ (L2CAP disconnect success)}
\end{align*}
\]

If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection, the IUT returns the following AVCT_DisconnectReq output parameter values to the Upper Tester:

RSP = 0x0000 (Request accepted)

The IUT transmits to the Lower Tester the L2CAP_DisconnectReq containing DCID and SCID values matching those of the channel established for AVCTP.

4.2.3 Connection Channel Establishment Response

- **Test Purpose**
  To verify that the IUT (CT and TG) is able to establish an AVCTP connection upon a connection request by another device. Note that testing of L2CAP is out of scope.

- **Test Case ID(s)**
  
  AVCTP/CT/CCM/BV-03-C  
  AVCTP/TG/CCM/BV-03-C

- **Reference**
  [3] 11.1

- **Initial Condition**
  - No L2CAP channel with the AVCTP PSM has yet been set-up between the IUT and the test system.
  - No limitations apply to the configuration of the L2CAP channel.
  - The PID identifying the profile using AVCTP is defined as PID_{Test_System}.
  - No profile with a PID = PID_{Test_System} is yet registered in the IUT.
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR_{Lower_Tester}.

- **Test Procedure**
  1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration, the Upper Tester registers the function ConnectInd_CB_{Test_System} for callback on the AVCT_Connect_Ind event by sending an AVCT_EventRegistration command to the IUT with the following parameter values:
Event = AVCT_Connect_Ind
Callback = ConnectInd_CB\text{Test System}
PID = PID_{Test System}

2. The Lower Tester sends an L2CAP_ConnectReq to the IUT and waits for the response.
3. The Lower Tester sends an L2CAP_ConfigReq to the IUT and waits for the response.
4. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection, upon a call to its callback function ConnectInd_CB\text{Test System} the Upper Tester sends an AVCT_ConnectRsp message to the IUT with the following parameter values:
   - BD_ADDR = BD_ADDR_{Lower Tester}
   - Connect Result = Valid value for L2CAP connect response result.
   - Status = Valid value for L2CAP connect response status.

   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for connection then the profile using the IUT issues an appropriate L2CAP_ConfigRsp to the Lower Tester.

**Figure 4.3: AVCTP/CT/CCM/BV-03-C & AVCTP/TG/CCM/BV-03-C**

- **Expected Outcome**
  
  **Pass verdict:**

  If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration:

  The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

  Result = 0x0000 (Event successfully registered)
The IUT calls the ConnectInd_CB function in the Upper Tester with the following parameter values:

\[ \text{BD_ADDR} = \text{BD_ADDR}_{\text{Lower_Tester}} \]

After reception of any expected AVCT_EventRegistration command from the Upper Tester and the L2CAP_ConnectReq from the Lower Tester, the IUT issues an L2CAP_ConnectRsp to the Lower Tester.

An L2CAP channel with the AVCTP PSM is set up between the IUT and the test system.

After reception of the L2CAP_ConfigReq from the Lower Tester, the IUT issues an appropriate L2CAP_ConfigRsp packet to the Lower Tester.

### 4.2.4 Disconnect Response

**Test Purpose**

To verify that the IUT (CT and TG) is able to disconnect an AVCTP connection upon a disconnect request from another device.

**Test Case ID(s)**

- AVCTP/CT/CCM/BV-04-C
- AVCTP/TG/CCM/BV-04-C

**Reference**

[3] 11.1

**Initial Condition**

- An L2CAP channel with the AVCTP PSM has been set-up between the IUT and the test system.
- The PID identifying the profile using AVCTP is defined as PID_{Test_System}.
- Only one profile with PID = PID_{Test_System} is registered in the IUT (e.g. Error! Reference source not found. or Error! Reference source not found. has been successfully executed).
- The BD_ADDR of the Lower Tester is defined as BD_ADDR_{Lower_Tester}.

**Test Procedure**

1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection event registration, the Upper Tester registers the DisconnectInd_CB function for callback on the AVCT_Disconnect_Ind event by sending an AVCT_EventRegistration command to the IUT with the following parameter values

   Event = AVCT_Disconnect_Ind
   Callback = DisconnectInd_CB_{Test_System}
   PID = PID_{Test_System}

2. The Lower Tester sends an L2CAP_DisconnectReq to the IUT and waits for the response.
One ACL link exists between the IUT and the Test System. An L2CAP channel with AVCTP PSM has been set up between the IUT and the Test System.

• Expected Outcome

Pass verdict:

If the Upper Tester and IUT conform to the AVCTP Upper Interface for disconnection event registration:

The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

Result = 0x0000 (Event successfully registered)

The IUT calls the DisconnectInd_CB_Test_System function in the Upper Tester with the following parameter values:

\[ \text{BD_ADDR} = \text{BD_ADDR}_{\text{Lower Tester}} \]

After reception of the L2CAP_DisconnectReq from the Lower Tester, the IUT issues an appropriate L2CAP_DisconnectRsp to the Lower Tester.
4.2.5 AVCTP/CT/CCM/BI-01-C [Connection Channel Establishment - AVCTP Connection Already Present]

• Test Purpose
To verify that the IUT (CT) does not establish a new L2CAP channel in attempts to connect an AVCTP channel when an AVCTP channel is already connected between the two devices. Note that testing of L2CAP is out of scope.

• Reference
[3] 11.2.1

• Initial Condition
- No L2CAP channel with the AVCTP PSM has yet been set-up between the IUT and the test system.
- Local side configuration settings: L2CAP configuration options.
- No limitations apply to the configuration of the L2CAP channel.
- The PID identifying the profile using AVCTP is defined as PID_{Test\_System}.
- The BD_ADDR of the Lower Tester is defined as BD_ADDR_{Lower\_Tester}.

• Test Procedure
1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration, the Upper Tester registers the ConnectCfm_CB_{Test\_System} function for callback on the AVCT_Connect_Cfm event by sending an AVCT_EventRegistration command to the IUT with the following parameter values:
   Event = AVCT_Connect_Cfm
   Callback = ConnectCfm_CB_{Test\_System}
   PID = PID_{Test\_System}
   After sending this command the Upper Tester waits on the response from the IUT.

2. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection, the Upper Tester sends an AVCT_ConnectReq command to the IUT with the following parameter values:
   BD_ADDR = BD_ADDR_{Lower\_Tester}
   PID = PID_{Test\_System}
   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for connection then the profile using the IUT initiates the L2CAP channel establishment with the Lower Tester by conducting Error! Reference source not found. The Upper Tester waits on a suitable response to the connect request.

3. If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection, the Upper Tester sends an AVCT_ConnectReq command to the IUT with the following parameter values:
   BD_ADDR = BD_ADDR_{Lower\_Tester}
   PID = PID_{Test\_System}
   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for connection then the profile using the IUT attempts to initiate another L2CAP connection via AVCTP using the same profile identification as step 2.
One ACL link exists between the IUT and the Test System.
No L2CAP channel exists between the IUT and the Test System.

**Figure 4.5: AVCTP/CT/CCM/BI-01-C**

- **Expected Outcome**

  **Pass verdict:**

  If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection event registration:

  The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:
  
  Result = 0x0000 (Event successfully registered)

  The IUT calls the ConnectCfm_CBTest_System function in the Upper Tester with the following parameters:

  BD_ADDR = BD_ADDR_Lower_Tester
Connect Result = 0x0000 (L2CAP Connect Request successful)
Config Result = 0x0000 (L2CAP Configure successful)
Status = L2CAP Connect Request Status

If the Upper Tester and IUT conform to the AVCTP Upper Interface for connection:

The IUT returns the following output parameter values to the Upper Tester on the first AVCT_ConnectReq:

RSP = 0x0000 (Request accepted)

The IUT returns the following output parameter values to the Upper Tester on the second AVCT_ConnectReq:

RSP = In the range 0x0001 to 0xFFFF (Request rejected)

The IUT does not establish a second L2CAP channel to the test system with the AVCTP PSM.

4.3 Non-Fragmented Messages

Test Group Objective:

- To verify the correct handling of non-fragmented messages on both CT and TG sides.

4.3.1 AVCTP/CT/NFR/BV-01-C [CT Sends Non-Fragmented Command Message]

- Test Purpose
To verify that the IUT (CT) formats correctly the following fields of the AVCTP command message:
"transaction label", "message type(C)", PID, "packet type", IPID and "message information".

- Reference
[3] 4.2, 4.3, 4.4, 4.5, 6.1.1, 6.2

- Initial Condition
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
  - The PID identifying the profile using AVCTP is defined as PID_{Test System}.
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR_{Test System}.

- Test Procedure
If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the Upper Tester sends an AVCT_SendMessage command to the IUT with the following parameter values:

BD_ADDR = BD_ADDR_{Test System}
Transaction = TRANS_{Test System}
Type = CR_{Test System} = 0 (Command Message)
PID = PID_{Test\_System}

Data = ADDRESS_{data\_buffer} (Buffer holding DATA[\_Upper\_Tester])

Length = L\_DATA[\_Upper\_Tester] <= MTU – 3 bytes

If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT sends an AVCTP packet to the Lower Tester.

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending:

- Expected Outcome

  **Pass verdict:**

  The IUT returns the following AVCT\_SendMessage output parameter values to the Upper Tester:

  Result = 0x0000 (Request accepted)
The AVCTP packet received by the Lower Tester has the following fields:

Transaction label = TRANS\text{Test\_System}

The Lower Tester receives an AVCTP packet with the following fields:

Packet\_type = 00 (Non-fragmented Message)

C/R = CR\text{Test\_System} = 0 (Command Message)

IPID = 0 (Command Message)

PID = PID\text{Test\_System}

Data = DATA\text{[Upper\_Tester\_or\_data\_supplied\_by\_the\_profile\_using\_the\_IUT]}

The message data has the length $L_{DATA\text{[Upper\_Tester\_or\_data\_supplied\_by\_the\_profile\_using\_the\_IUT]}}$ that matches the length of the data sent from the IUT.

### 4.3.2 AVCTP/TG/NFR/BV-02-C [TG Sends Non-Fragmented AVCTP Response Message]

- **Test Purpose**

  To verify that the IUT (TG) formats correctly the following fields of the AVCTP response message: "transaction label", "message type(R)", PID, "packet type", IPID and "message information".

- **Reference**

  [3] 4.2, 4.3, 4.4, 4.5, 6.1.1, 6.2

- **Initial Condition**

  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
  
  - The PID identifying the profile using AVCTP is defined as PID\text{Test\_System}.
  
  - The BD_ADDR of the Lower Tester is defined as BD\_ADDR\text{Test\_System}.
  
  - If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester has registered the function MessageInd\_CB\text{Test\_System} for callback on the AVCT_MessageRec\_Ind event.

- **Test Procedure**

  1. The Lower Tester sends an AVCTP packet to the IUT with the following fields:

     Transaction label = TRANS\text{Test\_System}
     
     Packet\_type = 00 (Non-fragmented message)
     
     C/R = 0 (Command Message)
     
     IPID = 0 (Command Message)
     
     PID = PID\text{Test\_System}
     
     Data = DATA\text{[Lower\_Tester\_or\_if\_the\_Upper\_Tester\_or\_IUT\_does\_not\_support\_the\_AVCTP\_Upper\_Interface\_for\_message\_sending,\_a\_valid\_command\_that\_the\_profile\_using\_the\_IUT\_responds\_to.]}
2. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, upon notification from the IUT the Upper Tester sends a AVCT_SendMessage command to the IUT with the following parameter values:
   
   $\begin{align*}
   BD\_ADDR &= BD\_ADDR\_Test\_System \\
   Transaction &= TRANS\_Test\_System \\
   Type &= CR\_Test\_System = 1 \text{ (Response Message)} \\
   PID &= PID\_Test\_System \\
   Data &= ADDRESS\_data\_buffer \text{ (Buffer containing DATA[Upper\_Tester])} \\
   Length &= L\_DATA[Upper\_Tester] <= MTU - 3\text{bytes}
   \end{align*}$

If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT issues an appropriate AVCTP packet to the Lower Tester.

---

**Figure 4.7: AVCTP/TG/NFR/BV-02-C**

- **Expected Outcome**

  **Pass verdict:**

  If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration the MessageInd_CBTest_System function in the Upper Tester is called with the following parameters:
BD_ADDR = BDADDR_Lower_Tester

Transaction = TRANS_Test_System

Type = 0x01 (Command message)

Data = ADDRESS_data_buffer (Buffer containing DATA[]_Lower_Tester)

Length = L_DATA[]_Lower_Tester

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the IUT returns the following AVCT_SendMessage output parameters to the Upper Tester:

Result = 0x0000 (Request accepted)

The Lower Tester receives an AVCTP packet with the following fields:

Transaction label = TRANS_Test_System

Packet_type = 00 (Non-fragmented Message)

C/R = CR(Test_System) = 1 (Response Message)

IPID = 0 (Valid Profile)

PID = PID/Test_System

Data = DATA[]_Upper_Tester or the data supplied by the profile using the IUT

The packet data has the length L_DATA[]_Upper_Tester that matches the length of the data sent.

4.3.3 AVCTP/TG/NFR/BV-03-C [TG Receives Incoming command]

• Test Purpose

To verify that the IUT (TG) correctly reports the parameters of the incoming command.

• Reference

[3] 4.2, 4.3, 4.4, 4.5, 6.1.1, 6.2

• Initial Condition

- One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
- The PID identifying the profile using AVCTP is defined as PID_Test_System.
- The BD_ADDR of the Lower Tester is defined as BD_ADDR_Test_System.

• Test Procedure

1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester registers the function MessageInd_CB_Test_System for callback on the AVCT_MessageRec_Ind event by sending an AVCT_EventRegistration command to the IUT with the following parameter values:
Event = AVCT_MessageRec_Ind
Callback = MessageInd_CBTest_System
PID = PIDTest_System

2. The Lower Tester issues an AVCTP packet to the IUT with the following fields:
   Transaction label = TRANSTest_System
   Packet_type = 00 (Non-fragmented Message)
   C/R = 0 (Command Message)
   IPID = 0 (Command Message)
   PID = PIDTest_System
   Data = DATA[Lower_Tester (data field of length L_DATA[Lower_Tester)]

   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message event registration the Lower Tester sends an unfragmented command that can be interpreted by the profile using the IUT.

   ![Diagram](image)

   **Figure 4.8: AVCTP/TG/NFR/BV-03-C**

   - **Expected Outcome**
   - **Pass verdict:**
When the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration:

The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

Result = 0x0000 (Event successfully registered)

The IUT calls the MessageInd_CB\textsubscript{Test\_System} callback function of the test system with the following parameters:

\begin{align*}
\text{BD\_ADDR} &= \text{BD\_ADDR}\textsubscript{Test\_System} \\
\text{Transaction} &= \text{TRANS}\textsubscript{Test\_System} \\
\text{Type} &= 0 \\
\text{Data} &= \text{DATA}[\]\textsubscript{Lower\_Tester} \\
\text{Length} &= L_{\text{DATA}[\]\textsubscript{Lower\_Tester}}
\end{align*}

When the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT correctly reacts on the incoming command from the Lower Tester.

4.3.4 AVCTP/CT/NFR/BV-04-C [CT Sends Non-Fragmented Command]

- **Test Purpose**
  To verify that the IUT (CT) reports correctly in the input parameters of the registered callback function the test system's BD\_ADDR and the following fields of an unfragmented response message: “transaction label”, “packet type” and “message information”.

- **Reference**
  [3] 11.1

- **Initial Condition**
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
  - The PID identifying the profile using AVCTP is defined as PID\textsubscript{Test\_System}.
  - The BD\_ADDR of the Lower Tester is defined as BD\_ADDR\textsubscript{Test\_System}.

- **Test Procedure**
  1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester registers the function MessageInd_CB\textsubscript{Test\_System} for callback on the AVCT\_MessageRec\_Ind event by sending an AVCT\_EventRegistration command to the IUT with the following parameter values:
     \begin{align*}
     \text{Event} &= \text{AVCT\_MessageRec\_Ind} \\
     \text{Callback} &= \text{MessageInd\_CB}\textsubscript{Test\_System} \\
     \text{PID} &= \text{PID}\textsubscript{Test\_System}
     \end{align*}
     After sending this command the Upper Tester waits on a response.
2. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the Upper Tester sends an AVCT_SendMessage command to the IUT with the following parameter values:

- **BD_ADDR** = BD_ADDR\textsubscript{Test System}
- **Transaction** = TRANS\textsubscript{Test System}
- **Type** = CR\textsubscript{Test System} = 0 (Command Message)
- **PID** = PID\textsubscript{Test System}
- **Data** = ADDRESSdata\textsubscript{buffer} (Buffer containing DATA[]\textsubscript{Upper Tester})
- **Length** = L\textsubscript{DATA[Upper Tester]} <= MTU - 3 bytes

If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT sends an AVCTP packet to the Lower Tester.

3. Upon receipt of the AVCTP command packet the Lower Tester issues an appropriate AVCTP response packet to the IUT with the following fields:

- **Transaction label** = TRANS\textsubscript{Test System}
- **Packet_type** = 00 (Non-fragmented Message)
- **C/R** = 1 (Response Message)
- **IPID** = 0 (Valid Profile)
- **PID** = PID\textsubscript{Test System}
- **Data** = DATA[]\textsubscript{Lower Tester} (data field of length L\textsubscript{DATA[Lower Tester]}.}
One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.

AVCTP packet
(Transaction label, Packet_type, C/R, IPID, PID, Data)

AVCT_SendMessage
(Result)

AVCT_EventRegistration
(Event, Callback, PID)

AVCT_EventRegistration
(Result)

AVCT_SendMessage
(BD_ADDR, Transaction, Type, PID, Data, Length)

AVCT_SendMessage
(Result)

MessageInd_CB\textsuperscript{Test System}
(BD_ADDR, Transaction, Type, Data, Length)

Figure 4.9: AVCTP/CT/NFR/BV-04-C

- Expected Outcome

Pass verdict:

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration:

The IUT returns the following AVCT_EventRegistration output parameters to the Upper Tester:

Result = 0x0000 (Event successfully registered)

The IUT calls the MessageInd\textsubscript{CB\textsuperscript{Test System}} callback function of the test system with the following parameters:

\begin{align*}
\text{BD_ADDR} & = \text{BD_ADDR}^{\text{Test System}} \\
\text{Transaction} & = \text{TRANS}^{\text{Test System}} \\
\text{Type} & = 0x01 \text{ (Command)}
\end{align*}
Data = DATA[Lower_Tester]
Length = LDATA[Lower_Tester]

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the IUT returns the following AVCT_SendMessage output parameters:

Result = 0x0000 (Request accepted)

The Lower Tester receives an AVCTP packet with the following fields:

Transaction label = TRANS[Test_System]
Packet_type = 00 (Non-fragmented Message)
C/R = C[R][Test_System] = 0 (Command Message)
IPID = 0 (Command Message)
PID = PID[Test_System]
Data = DATA[Upper_Tester] or data supplied by the profile using the IUT

The packet data has the length LDATA[Upper_Tester] that matches the length of the data sent.

4.3.5 AVCTP/TG/NFR/BI-01-C [TG Handles Control Message with Invalid PID]

- **Test Purpose**
  To verify that the IUT (TG) reports to the test system (CT) the reception of a control message intended for an invalid PID (PID not registered for reception of messages).

- **Reference**
  [3] 4.4, 6.1.1

- **Initial Condition**
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
  - The PID corresponding to a non-existent profile is defined as PIDInvalid.
  - No profile with a PID = PIDInvalid is registered in the IUT for the reception of messages.

- **Test Procedure**
  1. The Lower Tester sends an AVCTP packet with the following fields:
     - Transaction label = TRANS[Test_System]
     - Packet_type = 00 (Non-fragmented Message)
     - C/R = 0 (Command Message)
     - IPID = 0 (Command Message)
     - PID = PIDInvalid
     - Data = DATA[Lower_Tester] (data field of length LDATA[Lower_Tester]).
One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system. No profile with a PID = PIDInvalid is registered in the IUT for the reception of messages.

**Figure 4.10: AVCTP/TG/NFR/BI-01-C**

- **Expected Outcome**

  **Pass verdict:**
  
  The Lower Tester receives an AVCTP packet with the following data fields:

  - **Transaction label** = TRANSTest_System
  - **Packet_type** = 00 (Non-fragmented Message)
  - **C/R** = 1 (Response Message)
  - **IPID** = 1 (Invalid Profile)
  - **PID** = PIDInvalid (as assigned by the Lower Tester)

  No message information field is present.

### 4.4 Fragmented Messages

**Test Group Objective:**

- To verify the correct handling of fragmented messages on both CT and TG sides.

#### 4.4.1 AVCTP/CT/FRA/BV-01-C [CT Fragments Command Messages]

- **Test Purpose**

  To verify that the IUT (CT) fragments the command messages that cannot fit in a single L2CAP packet.

- **Reference**

  [3] 4.3, 6.1.2

- **Initial Condition**
- One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
- For the Lower Tester the value of the MTU for the L2CAP channel is $MTULowerTester$ bytes.
- The PID identifying the profile using AVCTP is defined as $PID_{Test\_System}$.
- The BD_ADDR of the Lower Tester is defined as $BD\_ADDR_{Test\_System}$.

**Test Procedure**

1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the Upper Tester sends an AVCTP_SendMessage command to the IUT with the following parameter values:
   
   \[
   \begin{align*}
   BD\_ ADDR &= BD\_ ADDR_{Test\_ System} \\
   Transaction &= TRANS_{Test\_ System} \\
   Type &= 1 \\
   PID &= PID_{Test\_ System} \\
   Data &= ADDRESS_{data\_ buffer} \ (Buffer\ holding\ DATA_{Upper\_ Tester}) \\
   Length &= LDATA_{Upper\_ Tester} = 3\times MTULowerTester - 6\text{ bytes}
   \end{align*}
   \]

   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT requests that a command message, of a length requiring three AVCTP packets, be sent to the IUT. Each AVCTP packet has a size equal to the L2CAP MTU limit.

**Figure 4.11: AVCTP/CT/FRA/BV-01-C**

**Expected Outcome**

**Pass verdict:**

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending the IUT returns the following AVCTP_SendMessage output parameters:
The IUT sends three AVCTP packets to the Lower Tester, each with a size equal to the L2CAP MTU limit.

AVCTP start packet parameters:
- Transaction = TRANS
- Packet_type = 01 (Start Packet)
- C/R = 0 (Command Message)
- IPID = 0 (Command Message)
- NoP = 3
- PID = PID
- Data = First part of the message information (length = MTU - 4)

AVCTP continuation packet parameters:
- Transaction = TRANS
- Packet_type = 10 (Continue Packet)
- C/R = 0 (Command Message)
- RFA = 0
- Data = Second part of the message information (length = MTU - 1)

AVCTP end packet parameters:
- Transaction = TRANS
- Packet_type = 11 (End Packet)
- C/R = 0 (Command Message)
- RFA = 0
- Data = Third part of the message information (length = MTU - 1).

4.4.2 AVCTP/TG/FRA/BV-02-C [TG Fragments Response Messages]
- Test Purpose
  To verify that the IUT (TG) fragments the response messages that cannot fit in a single L2CAP packet.
- Reference
  [3] 4.3, 6.1.2
• Initial Condition
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the test system.
  - For the Lower Tester the value of the MTU for the L2CAP channel is $MTU_{LowerTester}$ bytes.
  - The PID identifying the profile using AVCTP is defined as $PID_{Test\_System}$.
  - The BD_ADDR of the Lower Tester is defined as $BD\_ADDR_{Test\_System}$.
  - If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester has registered the function $MessageInd\_CB_{Test\_System}$ for callback on the $AVCT\_MessageRec\_Ind$ event.

• Test Procedure
  1. The Lower Tester issues an AVCTP packet to the IUT with the following fields:
     - Transaction label = $TRANS_{Test\_System}$
     - Packet type = 00 (Non-fragmented Message)
     - C/R = 0 (Command Message)
     - IPID = 0 (Command Message)
     - PID = $PID_{Test\_System}$
     - Data = $DATA[\_Lower\_Tester]$ (data field of length $L_{DATA[\_Lower\_Tester]}$)
     - The message has a data length of $L_{DATA[\_Lower\_Tester]} = MTU_{Lower\_Tester} - 3$ bytes.
  2. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, upon receipt of appropriate notification from the IUT the Upper Tester sends an $AVCT\_SendMessage$ command to the IUT with the parameter values:
     - $BD\_ADDR = BD\_ADDR_{Test\_System}$
     - Transaction = $TRANS_{Test\_System}$
     - Type = 0x02 (Response Message)
     - PID = $PID_{Test\_System}$
     - Data = $ADDRESS\_data\_buffer$ (Buffer holding $DATA[\_Upper\_Tester]$)
     - Length = $L_{DATA[\_Upper\_Tester]} = 3 \times MTU_{Lower\_Tester} - 6$ bytes
     - If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, upon receipt of the AVCTP packet from the Lower Tester the profile using the IUT submits response data to AVCTP of length $3 \times MTU_{Lower\_Tester} - 6$ bytes.
One ACL link exists between the IUT and the Test System.
One L2CAP channel exists between the IUT and the Test System.
The AVCTP_MessageReqInd event callback function MessageInd_CB_{TestSystem} has been registered.

**Expected Outcome**

**Pass verdict:**

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the IUT calls the MessageInd_CB_{TestSystem} callback function of the test system with the following parameters:

- BD_ADDR = BD_ADDR_{TestSystem}
- Transaction = TRANS_{TestSystem}
- Type = 0x01 (Command Message)
- Data = ADDRESS_{data_buffer} (Buffer holding DATA[]_{LowerTester})
- Length = L\_DATA\_\_LowerTester

If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the IUT returns the following AVCT_SendMessage output parameters:

- Result = 0x0000 (Request accepted)

The IUT sends three AVCTP packets to the Lower Tester, each with a size equal to the L2CAP MTU_{LowerTester} limit.
AVCTP start packet:

Transaction = TRANS\textsuperscript{Test\_System}
Packet\_type = 01 (Start Packet)
C/R = 1 (Response Message)
IPID = 0 (Valid Profile)
NoP = 3
PID = PID\textsuperscript{Test\_System}
Data = First part of the message information (length = MTU\textsubscript{Lower\_Tester} - 4)

AVCTP continuation packet:

Transaction = TRANS\textsuperscript{Test\_System}
Packet\_type = 10 (Continue Packet)
C/R = 1 (Response Message)
RFA = 0
Data = Second part of the message information (length = MTU\textsubscript{Lower\_Tester} - 1)

AVCTP end packet:

Transaction = TRANS\textsuperscript{Test\_System}
Packet\_type = 11 (End Packet)
C/R = 1 (Response Message)
RFA = 0
Data = Third part of the message information (length = MTU\textsubscript{Lower\_Tester} - 1).

4.4.3 AVCTP/TG/FRA/BV-03-C [TG Reassembles Fragmented Command Message]

- **Test Purpose**
  To verify that the IUT (TG) reassembles a fragmented command message.

- **Reference**
  [3] 4.3, 6.1.2

- **Initial Condition**
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the Lower Tester.
- The PID identifying the profile using AVCTP is defined as PID_{Test\_System}.
- The BD\_ADDR of the Lower Tester is defined as BD\_ADDR_{Test\_System}.
- If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester has registered the function MessageInd\_CB_{Test\_System} for callback on the AVCT_MessageRec\_Ind event.

Testing Procedure

1. The Lower Tester issues an AVCTP message whose information is fragmented into three AVCTP packets with the following fields:

   **AVCTP start packet:**
   - Transaction = TRANS_{Test\_System}
   - Packet_type = 01 (Start Packet)
   - C/R = 0 (Command Message)
   - IPID = 0 (Command Message)
   - NoP = 3
   - PID = PID_{Test\_System}
   - Data = First part of the message information

   **AVCTP continuation packet:**
   - Transaction = TRANS_{Test\_System}
   - Packet_type = 10 (Continue Packet)
   - C/R = 0 (Command Message)
   - RFA = 0
   - Data = Second part of the message information

   **AVCTP end packet:**
   - Transaction = TRANS_{Test\_System}
   - Packet_type = 11 (End Packet)
   - C/R = 0 (Command Message)
   - RFA = 0
   - Data = Third part of the message information

   If the Upper Tester and IUT do not conform to the AVDTP Upper Interface for message event registration then the Lower Tester sends a command that can be interpreted by the profile using the IUT.
One ACL link exists between the IUT and the Test System.
One L2CAP channel exists between the IUT and the Test System.
The AVCTP_MessageReqInd event callback function MessageInd_CB_Testsyste has been registered.

AVCTP packet
(Transaction, Packet Type, C/R, PIID, NoP, PID, Data)

AVCTP packet
(Transaction, Packet Type, C/R, RFA, Data)

AVCTP packet
(Transaction, Packet Type, C/R, RFA, Data)

MessagInd_CB_Testsyste
(BD_ADDR, Transaction, Type, Data, Length)

Figure 4.13: AVCTP/TG/FRA/BV-03-C

- Expected Outcome
  
  Pass verdict:
  When the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the IUT receives three AVCTP packets from the Lower Tester, reassembles the message and calls the MessageInd_CB_Testsyste callback function with the following parameters:

  BD_ADDR       = BD_ADDR_Testsyste
  Transaction   = TRANS_Testsyste
  Type          = 0x01 (Command Message)
  Data          = ADDRESS_data_buffer (Buffer holding DATA[]Lower_Tester)
  Length        = L_DATA[]Lower_Tester

  When the Upper Tester or IUT does not conform to the AVCTP Upper Interface for message sending, the profile using the IUT correctly reacts to the incoming command from the Lower Tester.

- Notes
  The size of the fragmented packets does not need to be equal to the MTU size. The specification does not disallow sending of fragmented packets that are smaller than the MTU size.

4.4.4 AVCTP/CT/FRA/BV-04-C [CT Reassembles a Fragmented Response Message]

- Test Purpose
To verify that the IUT (CT) reassembles the fragmented response messages.

- Reference
  
  [3] 4.3, 6.1.2

- Initial Condition
  - One L2CAP channel with the AVCTP PSM is set-up and configured between the IUT and the Lower Tester.
  - The PID identifying the profile using AVCTP is defined as PID_{Test\_System}.
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR_{Test\_System}.
  - If the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the Upper Tester has registered the function MessageInd_CB_{Test\_System} for callback on the AVCT_MessageRec_Ind event.

- Test Procedure
  1. If the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the Upper Tester sends an AVCT_SendMessage command to the IUT with the following parameter values:

     | Parameter       | Value                                      |
     |-----------------|--------------------------------------------|
     | BD_ADDR         | BD_ADDR_{Test\_System}                     |
     | Transaction     | TRANS_{Test\_System}                       |
     | Type            | 0x01 (Command Message)                     |
     | PID             | PID_{Test\_System}                         |
     | Data            | ADDRESS_{data\_buffer} (Buffer containing DATA[Upper\_Tester]) |
     | Length          | L_{DATA[Upper\_Tester]}                    |

   If the Upper Tester or IUT do not conform to the AVCTP Upper Interface for message sending, the profile using the IUT sends an AVCTP command to the Lower Tester.

  2. Upon receipt of an AVCTP command, the IUT sends an AVCTP response message fragmented into 3 AVCTP packets with the following fields:

     AVCTP start packet:
     | Parameter       | Value                                      |
     |-----------------|--------------------------------------------|
     | Transaction     | TRANS_{Test\_System}                       |
     | Packet_type     | 01 (Start Packet)                         |
     | C/R             | 1 (Response Message)                      |
     | IPID            | 0 (Valid Profile)                         |
     | NoP             | 3                                          |
     | PID             | PID_{Test\_System}                         |
     | Data            | First part of DATA[Lower\_Tester]          |

     AVCTP continuation packet:
     | Parameter       | Value                                      |
     |-----------------|--------------------------------------------|
     | Transaction     | TRANS_{Test\_System}                       |
     | Packet_type     | 10 (Continue Packet)                       |
     | C/R             | 1 (Response message)                      |
     | RFA             | 0                                          |
     | Data            | Second part of DATA[Lower\_Tester]         |

     AVCTP end packet:
     | Parameter       | Value                                      |
     |-----------------|--------------------------------------------|
     | Transaction     | TRANS_{Test\_System}                       |
     | Packet_type     | 11 (End Packet)                           |
     | C/R             | 1 (Response Message)                      |
     | RFA             | 0                                          |
     | Data            | Third part of DATA[Lower\_Tester]          |
One ACL link exists between the IUT and the Test System.
One L2CAP channel exists between the IUT and the Test System.
The AVCTP_MessageReqInd event callback function MessageInd_CB_Test_System has been registered.

AVCTP packet
(Transaction, Packet Type, C/R, IPID, PID, Data)

AVCTP packet
(Transaction, Packet Type, C/R, IPID, NoP, PID, Data)

AVCTP packet
(Transaction, Packet Type, C/R, RFA, Data)

AVCTP packet
(Transaction, Packet Type, C/R, RFA, Data)

Figure 4.14: AVCTP/CT/FRA/BV-04-C

- Expected Outcome

Pass verdict:

When the Upper Tester and IUT conform to the AVCTP Upper Interface for message sending, the IUT returns the following AVCT_SendMessage output parameters:

Result = 0x0000 (Request accepted)

The IUT sends appropriate AVCTP command packet(s) to the Lower Tester.

When the Upper Tester and IUT conform to the AVCTP Upper Interface for message event registration, the IUT calls the MessageInd_CBTestSystem callback function with the following parameters:

BD_ADDR = BD_ADDR_{Test System}
Transaction = TRANS_{Test System}
Type = 0x02 (Response Message)
Data = ADDRESS_{data_buffer} (Buffer containing DATA[\_]_{Lower_Tester})
Length = L_{DATA[\_]_{Lower_Tester}}
When the Upper Tester does not conform to the AVCTP Upper Interface for message event registration, the profile using the IUT correctly reacts to the incoming response from the Lower Tester.

- **Notes**

  The size of the fragmented packets does not need to be equal to the MTU size. The specification does not disallow sending of fragmented packets that are smaller than the MTU size.
# 5 Test Case Mapping

The Test Case Mapping Table (TCMT) maps test cases to specific requirements in the ICS. The product shall be tested in all roles for which support is declared in the ICS document.

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 Audio/Video Control Transport Protocol (AVCTP) [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.

**Test Case Applicable:** May be used to note if a test is required based on the supported features.

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>
<th>Test Case Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVCTP 2/11</td>
<td>Connect Request</td>
<td>AVCTP/CT/CCM/BV-01-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/11</td>
<td>Connect Request</td>
<td>AVCTP/TG/CCM/BV-01-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/12</td>
<td>Disconnect Request</td>
<td>AVCTP/CT/CCM/BV-02-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/12</td>
<td>Disconnect Request</td>
<td>AVCTP/TG/CCM/BV-02-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/9</td>
<td>Event registration for connection request</td>
<td>AVCTP/CT/CCM/BV-03-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/9</td>
<td>Event registration for connection request</td>
<td>AVCTP/TG/CCM/BV-03-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/10</td>
<td>Event registration for disconnection request</td>
<td>AVCTP/CT/CCM/BV-04-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/10</td>
<td>Event registration for disconnection request</td>
<td>AVCTP/TG/CCM/BV-04-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/14</td>
<td>Support for multiple AVCTP channel establishment requests</td>
<td>AVCTP/CT/CCM/BI-01-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/13</td>
<td>Send non-fragmented message (CT)</td>
<td>AVCTP/CT/NFR/BV-01-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/13</td>
<td>Send non-fragmented message (TG)</td>
<td>AVCTP/TG/NFR/BV-02-C</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/2 OR AVCTP 3/3 OR AVCTP 3/4 OR AVCTP 3/5 OR AVCTP 3/6 OR AVCTP 3/7 OR AVCTP 3/8</td>
<td>Event registration for message reception (TG)</td>
<td>AVCTP/TG/NFR/BV-03-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/2 OR AVCTP 2/3 OR AVCTP 2/4 OR AVCTP 2/5 OR AVCTP 2/6 OR AVCTP 2/7 OR AVCTP 2/8</td>
<td>Event registration for message reception (CT)</td>
<td>AVCTP/CT/NFR/BV-04-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/1 AND AVCTP 2/13</td>
<td>Send fragmented message (CT)</td>
<td>AVCTP/CT/FRA/BV-01-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP (3/1 AND 3/13)</td>
<td>Send fragmented message (TG)</td>
<td>AVCTP/TG/FRA/BV-02-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 3/1</td>
<td>Fragmented message reassembly (TG)</td>
<td>AVCTP/TG/FRA/BV-03-C</td>
<td></td>
</tr>
<tr>
<td>AVCTP 2/1</td>
<td>Fragmented message reassembly (CT)</td>
<td>AVCTP/CT/FRA/BV-04-C</td>
<td></td>
</tr>
</tbody>
</table>

*Table 5.1: Test Case Mapping*