Audio/Video Control Transport Protocol (AVCTP)

***Bluetooth*® Test Suite**

- **Revision:** AVCTP.TS.p12
- **Revision Date:** 2024-07-01
- **Prepared By:** BTI
- **Published during TCRL:** TCRL.2024-1

Contents

1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and test cases to test the implementation of the Bluetooth Audio/Video Control Transport Protocol (AVCTP) with the objective to provide a high probability of air interface interoperability between the tested implementation and other manufacturers' Bluetooth devices.

2 References, definitions, and abbreviations

2.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [3], [5], and [6].

[1] OSI Conformance testing methodology and framework for protocol recommendations for ITU-T applications – General Concepts ITU-T Recommendation X.290 (04/95)

[2] OSI Conformance testing methodology and framework for protocol recommendations for ITU-T applications - Abstract Test Suite Specification ITU-T Recommendation X.291 (04/95)

[3] Audio/Video Control Transport Protocol Specification

[4] ICS Proforma for Audio/Video Control Transport Protocol

[5] Bluetooth Core Specification, Version 2.0 or later

[6] Test Strategy and Terminology Overview

2.2 Definitions

In this Bluetooth document, the definitions from [3], [5], and [6] apply.

2.3 Acronyms and abbreviations

In this Bluetooth document, the definitions, acronyms, and abbreviations from [3], [5], and [6] apply.

3 Test Suite Structure (TSS)

3.1 Overview

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

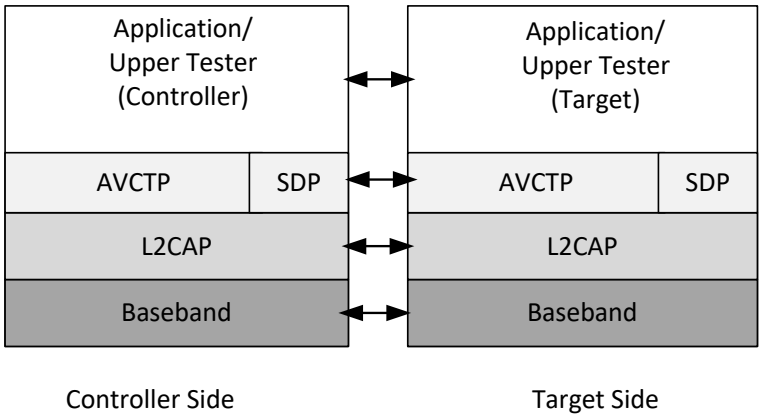

Figure 3.1: AVCTP test model

3.2 Test Strategy

The test objectives are to verify the functionality of the AVCTP implementation within a Bluetooth Host and enable interoperability between Bluetooth Hosts on different devices. The testing approach covers mandatory and optional requirements in the specification and matches these to the support of the IUT as described in the ICS. Any defined test herein is applicable to the IUT if the ICS logical expression defined in the Test Case Mapping Table (TCMT) evaluates to true. 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 test equipment provides an implementation of the Radio Controller and the parts of the Host needed to perform the test cases defined in this Test Suite. A Lower Tester acts as the IUT's peer device and interacts with the IUT over-the-air interface. The configuration, including the IUT, needs to implement similar capabilities to communicate with the test equipment. For some test cases, it is necessary to stimulate the IUT from an Upper Tester. In practice, this could be implemented as a special test interface, a Man Machine Interface (MMI), or another interface supported by the IUT.

The IUT, the Lower Tester, and the Upper Tester are referred to in the message sequence charts (MSCs) describing the test procedures in their respective test cases in Sections 4.2, 4.3, and 4.4. In testing, only one point-to-point configuration is considered where an ACL link has been set up between the IUT and the test system.

The IUT can be in either the CT or the TG role. The selected role is specified for each test case.

This Test Suite contains Valid Behavior (BV) tests complemented with Invalid Behavior (BI) tests where required. The test coverage mirrored in the Test Suite Structure is the result of a process that started with catalogued specification requirements that were logically grouped and assessed for testability enabling coverage in defined test purposes.

3.3 Test groups

The following test groups have been defined:

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

4 Test cases (TC)

4.1 Introduction

4.1.1 Test case identification conventions

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

Identifier Abbreviation	**Spec Identifier <spec abbreviation>**
AVCTP	Audio/Video Control Transport
Identifier Abbreviation	**Role Identifier <IUT role>**
CT	Controller role
TG	Target role
Identifier Abbreviation	**Feature Identifier <feat>**
CCM	Connection channel management
FRA	Fragmented message
NFR	Non-fragmented message

Table 4.1: AVCTP TC feature naming conventions

4.1.2 Conformance

When conformance is claimed for a particular specification, all capabilities are to be supported in the specified manner. The mandated tests from this Test Suite depend on the capabilities to which conformance is claimed.

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 not excluded by the specification
- That capabilities enabled by the implementations are sustained over durations expected by the use case
- That the implementation gracefully handles any quantity of data expected by the use case
- That in cases where more than one valid interpretation of the specification exists, the implementation complies with at least one interpretation and gracefully handles other interpretations
- That the implementation is immune to attempted security exploits

A single execution of each of the required tests is required to constitute a Pass verdict. However, it is noted that 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 generated by the Bluetooth SIG qualification tool, with 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 erratum request such that the issue may be addressed.

4.1.3 Pass/Fail verdict conventions

Each test case has an Expected Outcome section. The IUT is granted the Pass verdict when all the detailed pass criteria conditions within the Expected Outcome section are met.

The convention in this Test Suite is that, unless there is 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, then the outcome of the test is a Fail verdict.

4.2 Connection Channel Management

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

  Verify that the IUT (CT and TG) can request a connection channel establishment. 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 Lower Tester.
  - 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 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 Case Configuration

Test Case
AVCTP/CT/CCM/BV-01-C [Connection Channel Establishment Request]
AVCTP/TG/CCM/BV-01-C [Connection Channel Establishment Request]

Table 4.2: Connection Channel Establishment Request test cases

- Test Procedure

  1. If the Upper Tester and the IUT conform to the AVCTP Upper Interface for connection event registration, the Upper Tester registers the function $ConnectCfm\_CB_{Test\_System}$ for callback on the AVCT_Connect_Cfm event by sending an AVCT_EventRegistration command to the IUT with the following input parameter values:

     - 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 the 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:

   - BD_ADDR = $BD\_ADDR_{Lower\_Tester}$
   - PID = PID_{Test_System}

   If the Upper Tester or the IUT does 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.

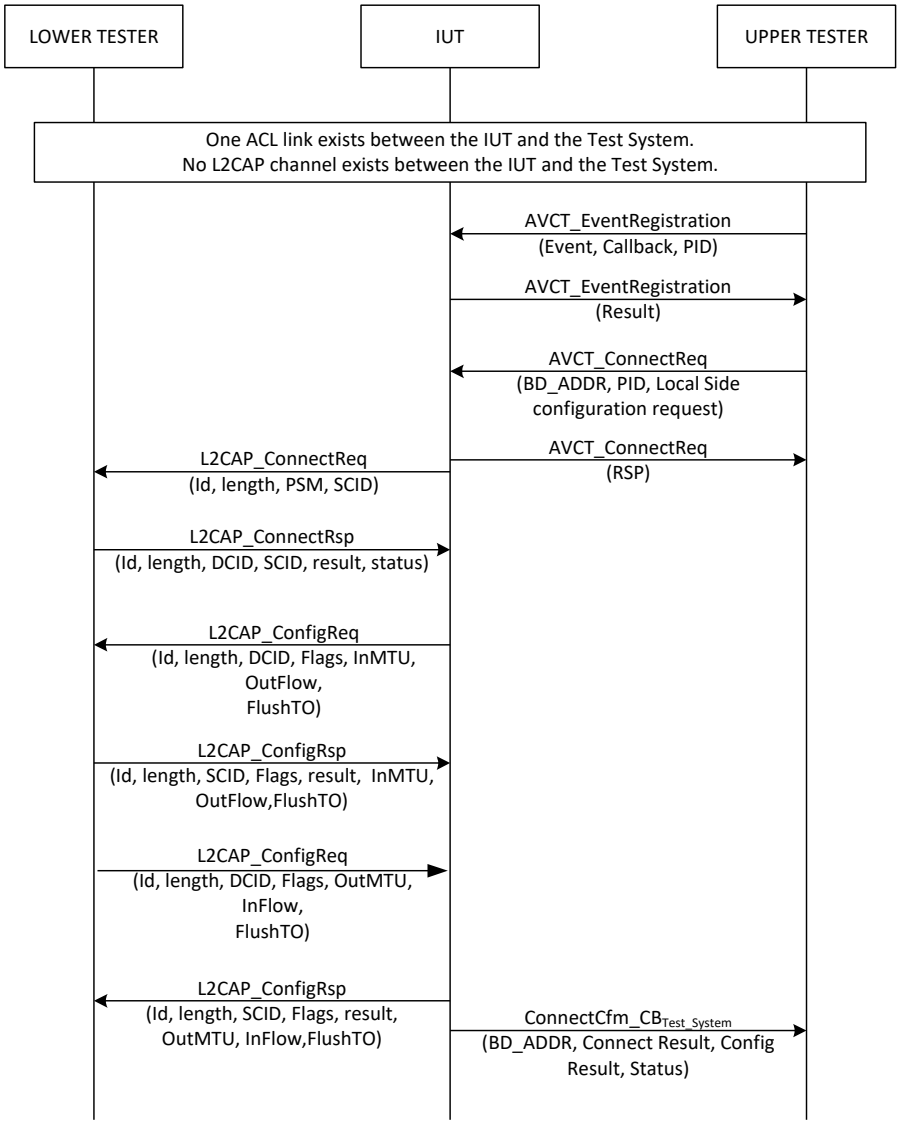

Figure 4.1: Connection Channel Establishment Request – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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$_{Test_System}$ 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 the 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

  Verify that the IUT (CT and TG) can disconnect a channel previously established.

- 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 PID$_{Test_System}$.
  - Only one profile with PID = PID$_{Test_System}$ is registered in the IUT (e.g., AVCTP/CT/CCM/BV-01-C [Connection Channel Establishment Request] or AVCTP/TG/CCM/BV-01-C [Connection Channel Establishment Request] has been successfully executed).
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR$_{Lower_Tester}$.

- Test Case Configuration

Test Case
AVCTP/CT/CCM/BV-02-C [Disconnect a Channel]
AVCTP/TG/CCM/BV-02-C [Disconnect a Channel]

Table 4.3: Disconnect a Channel test cases

- Test Procedure

1. If the Upper Tester and the IUT conform to the AVCTP Upper Interface for disconnection event registration, the Upper Tester registers the function DisconnectCfm_CB$_{Test_System}$ 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 the 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_$_{ADDRLower_Tester}$
    - PID = PID$_{Test_System}$

    If the Upper Tester or the IUT does 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.

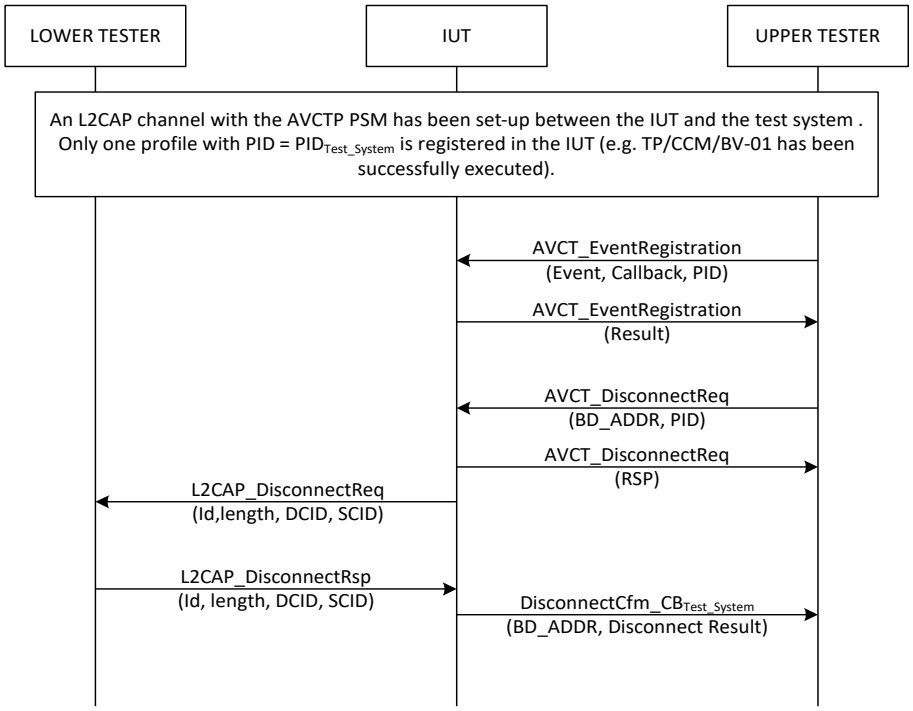

Figure 4.2: Disconnect a Channel – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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$_{Test_System}$ function in the Upper Tester with the following parameter values:

  - BD_ADDR = BD_ADDR$_{Lower_Tester}$
  - Disconnect Result = 0x0000 (L2CAP disconnect success)

  If the Upper Tester and the 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

  Verify that the IUT (CT and TG) can establish an AVCTP connection upon a connection request by another device. Note that testing of L2CAP is out of scope.

- 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 Case Configuration

Test Case
AVCTP/CT/CCM/BV-03-C [Connection Channel Establishment Response]
AVCTP/TG/CCM/BV-03-C [Connection Channel Establishment Response]

Table 4.4: Connection Channel Establishment Response test cases

- Test Procedure

1. If the Upper Tester and the 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$_{Test_System}$
    - 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 the IUT conform to the AVCTP Upper Interface for connection, upon a call to its callback function ConnectInd_CB$_{Test_System}$ 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 the IUT does not conform to the AVCTP Upper Interface for connection, then the profile using the IUT issues an appropriate L2CAP_ConfigRsp to the Lower Tester.

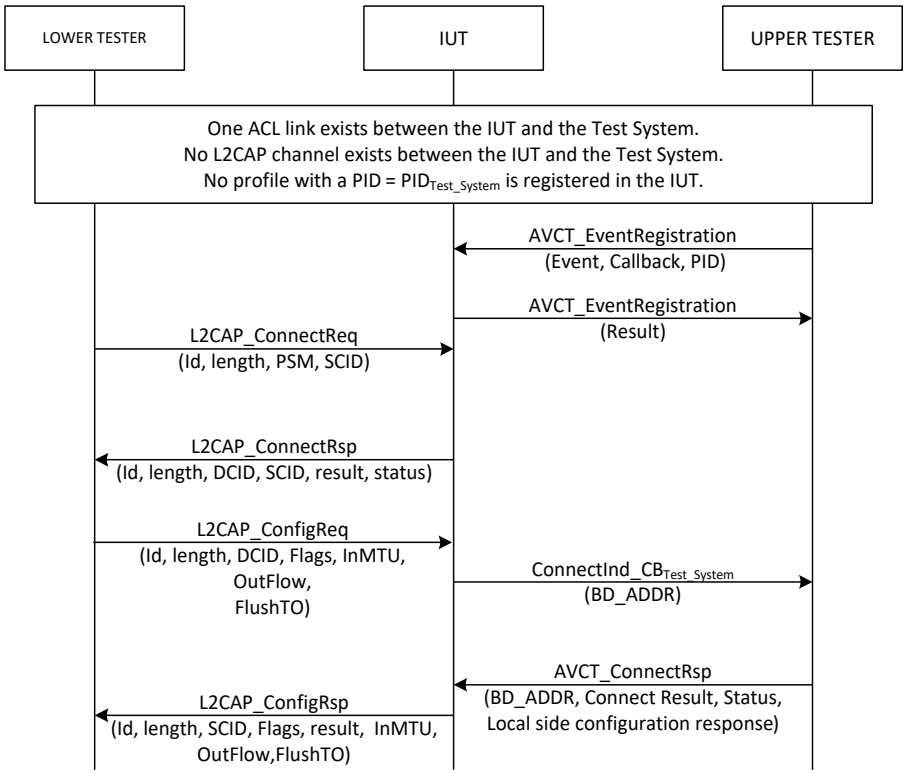

Figure 4.3: Connection Channel Establishment Response – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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$_{Test_System}$ function in the Upper Tester with the following parameter values:

  - BD_ADDR = BD_ADDR$_{Lower_Tester}$

  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

  Verify that the IUT (CT and TG) can disconnect an AVCTP connection upon a disconnect request from another device.

- 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., AVCTP/CT/CCM/BV-01-C [Connection Channel Establishment Request] or AVCTP/TG/CCM/BV-01-C [Connection Channel Establishment Request] has been successfully executed).
  - The BD_ADDR of the Lower Tester is defined as BD_ADDR$_{Lower_Tester}$.

- Test Case Configuration

Test Case
AVCTP/CT/CCM/BV-04-C [Disconnect Response]
AVCTP/TG/CCM/BV-04-C [Disconnect Response]

Table 4.5: Disconnect Response test cases

- Test Procedure

  1. If the Upper Tester and the IUT conform to the AVCTP Upper Interface for disconnection event registration, the Upper Tester registers the DisconnectInd_CB$_{Test_System}$ 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.

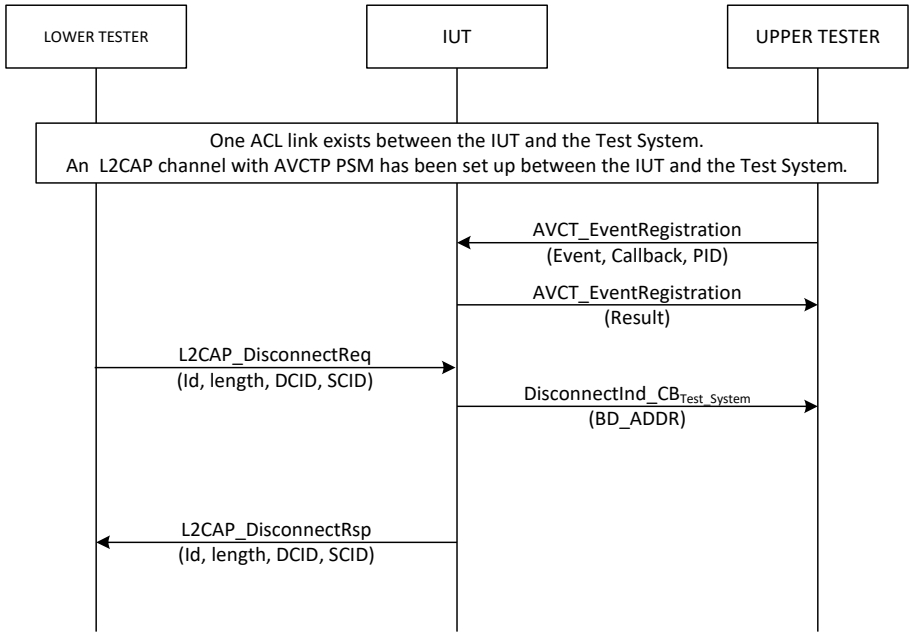

Figure 4.4: Disconnect Response – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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:

  - BD_ADDR = BD_ADDR$_{Lower_Tester}$

  After reception of the L2CAP_DisconnectReq from the Lower Tester, the IUT issues an appropriate L2CAP_DisconnectRsp to the Lower Tester.

AVCTP/CT/CCM/BI-01-C [Connection Channel Establishment – AVCTP Connection Already Present]

- Test Purpose

  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 the 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 the 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 the IUT does 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 AVCTP/CT/CCM/BV-01-C [Connection Channel Establishment Request].
     The Upper Tester waits on a suitable response to the connect request.
  3. If the Upper Tester and the 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 the IUT does 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.

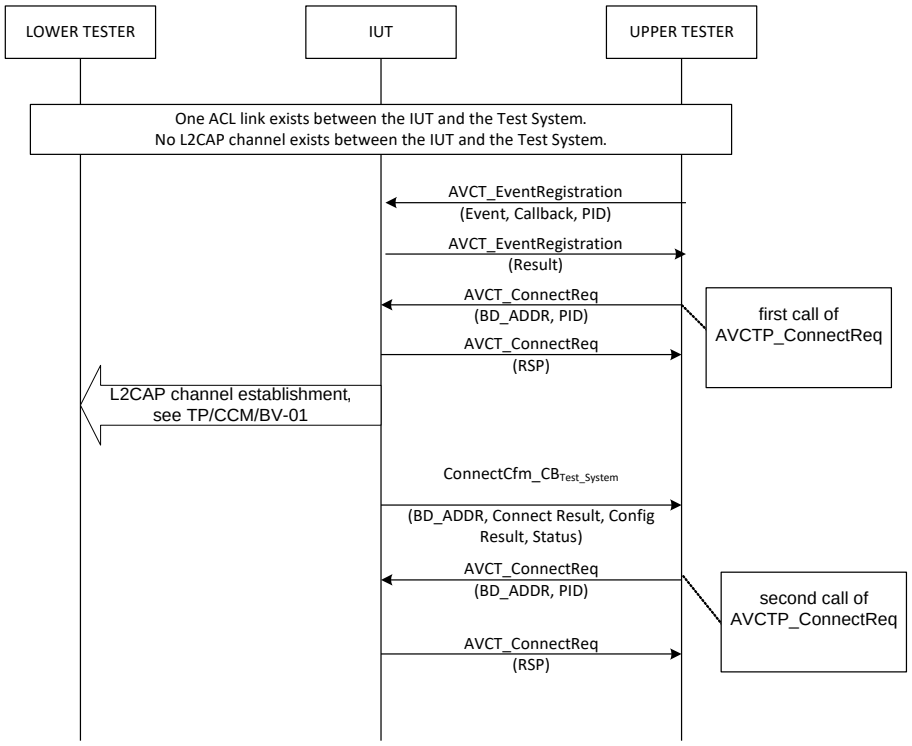

Figure 4.5: AVCTP/CT/CCM/BI-01-C [Connection Channel Establishment – AVCTP Connection Already Present] – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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$_{Test_System}$ 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 the 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

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

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

- Test Purpose

  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 the 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 – 3bytes

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

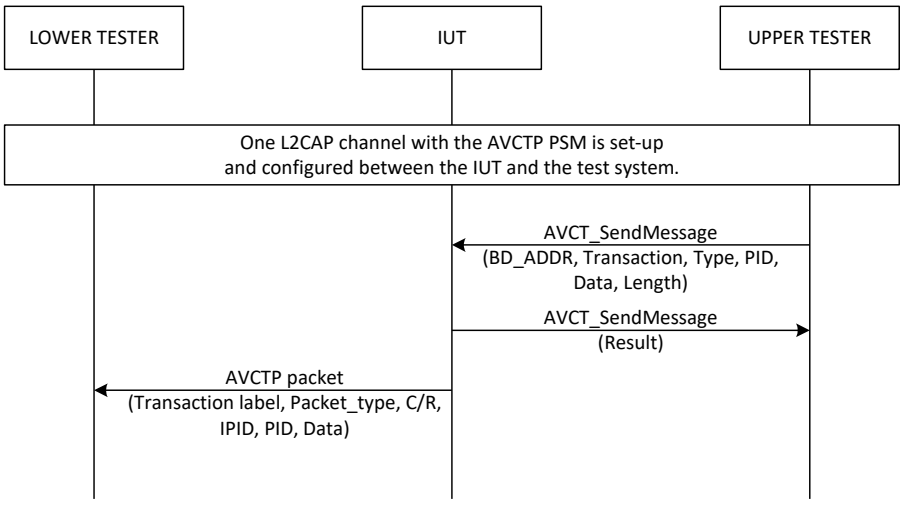

Figure 4.6: AVCTP/CT/NFR/BV-01-C [CT Sends Non-Fragmented Command Message] - MSC

- Expected Outcome

  Pass verdict

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

  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 = $TRANS_{Test\_System}$

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

  - Packet_type = 00 (Non-fragmented Message)
  - C/R = CR_{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 message data has the length $L_{DATA[]Upper\_Tester}$ that matches the length of the data sent from the IUT.

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

- Test Purpose

  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_{Test_System}.
  - The BD_ADDR of the Lower Tester is defined as $BD\_ADDR_{Test\_System}$.
  - If the Upper Tester and the 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 sends 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 = $DATA[]_{Lower\_Tester}$, or if the Upper Tester or the 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 the 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:

   - BD_ADDR = $BD\_ADDR_{Test\_System}$
   - Transaction = $TRANS_{Test\_System}$
   - Type = CR_{Test_System} = 1 (Response Message)
   - PID = PID_{Test_System}
   - Data = $ADDRESS_{data\_buffer}$ (Buffer containing $DATA[]_{Upper\_Tester}$)
   - Length = $L_{DATA[]Upper\_Tester}$ <= MTU – 3bytes

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

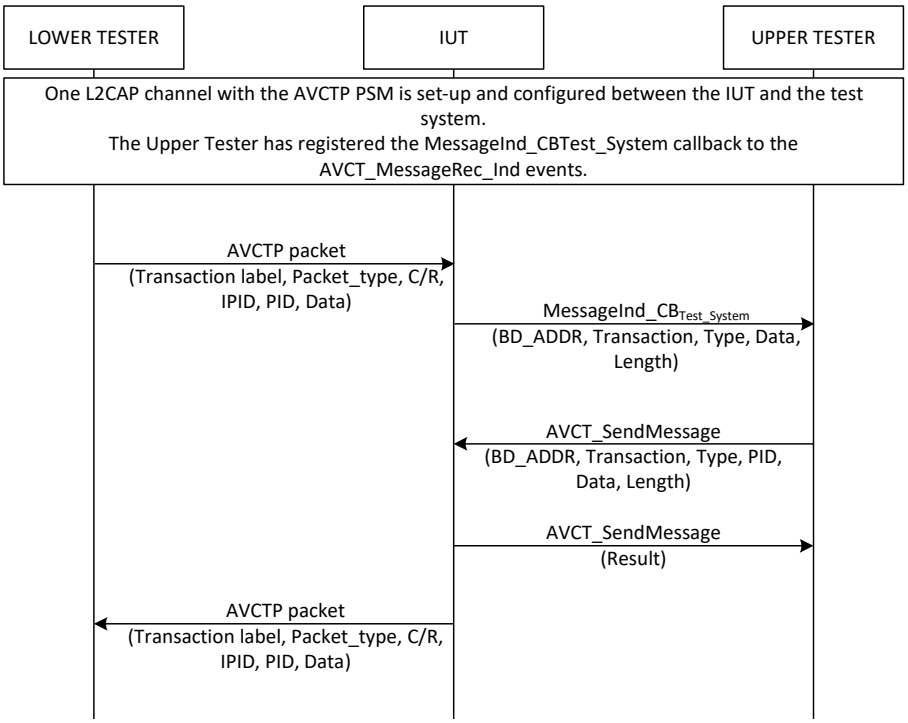

Figure 4.7: AVCTP/TG/NFR/BV-02-C [TG Sends Non-Fragmented AVCTP Response Message] – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the IUT conform to the AVCTP Upper Interface for message event registration, the $MessageInd\_CB_{Test\_System}$ function in the Upper Tester is called with the following parameters:

  - BD_ADDR = $BD\_ADDR_{Lower\_Tester}$
  - 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 the 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.

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

- Test Purpose

  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 the 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 = $MessageInd\_CB_{Test\_System}$
     - PID = PID_{Test_System}

  2. 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 $L_{DATA[]Lower\_Tester}$)

     If the Upper Tester or the IUT does 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.

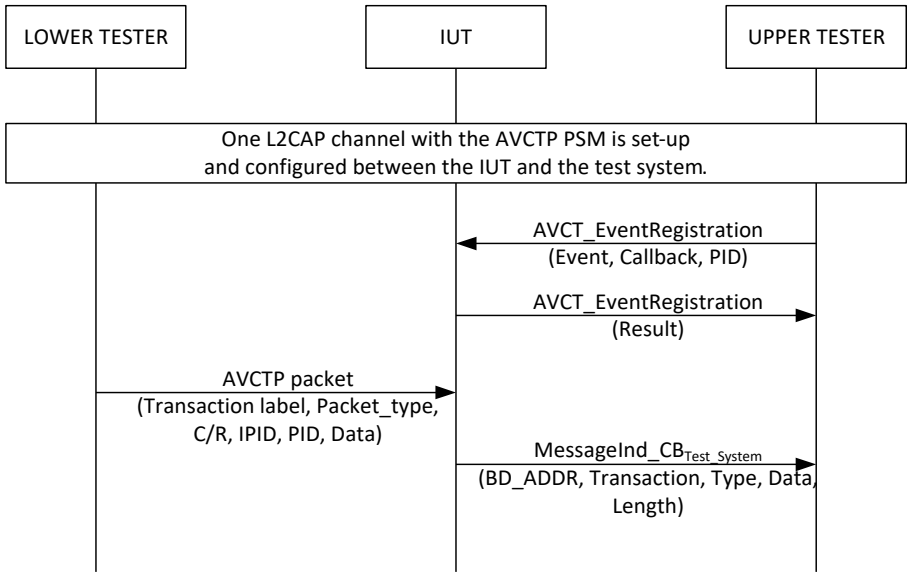

Figure 4.8: AVCTP/TG/NFR/BV-03-C [TG Receives Incoming Command] – MSC

- Expected Outcome

  Pass verdict

  When the Upper Tester and the 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$_{Test_System}$ callback function of the test system with the following parameters:

  - BD_ADDR = BD_ADDR$_{Test_System}$
  - Transaction = TRANS$_{Test_System}$
  - Type = 0
  - Data = DATA[]$_{Lower\_Tester}$
  - Length = L$_{DATA[]Lower_Tester}$

  When the Upper Tester or the IUT does 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.

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

- Test Purpose

  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_{Test_System}.
  - The BD_ADDR of the Lower Tester is defined as $BD\_ADDR_{Test\_System}$.
- Test Procedure
  1. If the Upper Tester and the 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 = $MessageInd\_CB_{Test\_System}$
     - PID = PID_{Test_System}

     After sending this command, the Upper Tester waits on a response.
  2. If the Upper Tester and the 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 containing $DATA[]_{Upper\_Tester}$)
     - Length = $L_{DATA[]Upper\_Tester}$ <= MTU - 3bytes

     If the Upper Tester or the IUT does 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 = $TRANS_{Test\_System}$
     - Packet_type = 00 (Non-fragmented Message)
     - C/R = 1 (Response Message)
     - IPID = 0 (Valid Profile)
     - PID = PID_{Test_System}
     - Data = $DATA[]_{Lower\_Tester}$ (data field of length $L_{DATA[]Lower\_Tester}$)

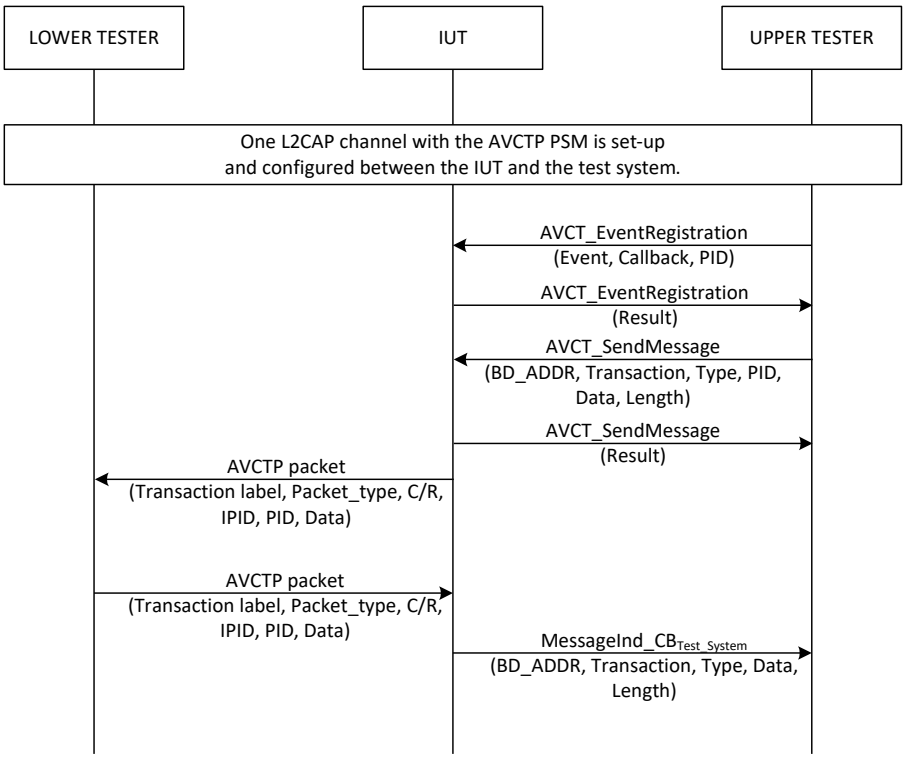

Figure 4.9: AVCTP/CT/NFR/BV-04-C [CT Sends Non-Fragmented Command] – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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$_{Test_System}$ callback function of the test system with the following parameters:

  - BD_ADDR = BD_ADDR$_{Test_System}$
  - Transaction = TRANS$_{Test_System}$
  - Type = 0x01 (Command)
  - Data = DATA[]$_{Lower\_Tester}$
  - Length = L$_{DATA[]Lower_Tester}$

  If the Upper Tester and the 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 = CR$_{Test_System}$ = 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 $L_{DATA[]Upper\_Tester}$ that matches the length of the data sent.

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

- Test Purpose

  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 $PID_{Invalid}$.
  - No profile with a PID = $PID_{Invalid}$ 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 = $PID_{Invalid}$
     - Data = $DATA[]_{Lower\_Tester}$ (data field of length $L_{DATA[\ ]Lower\_Tester}$)

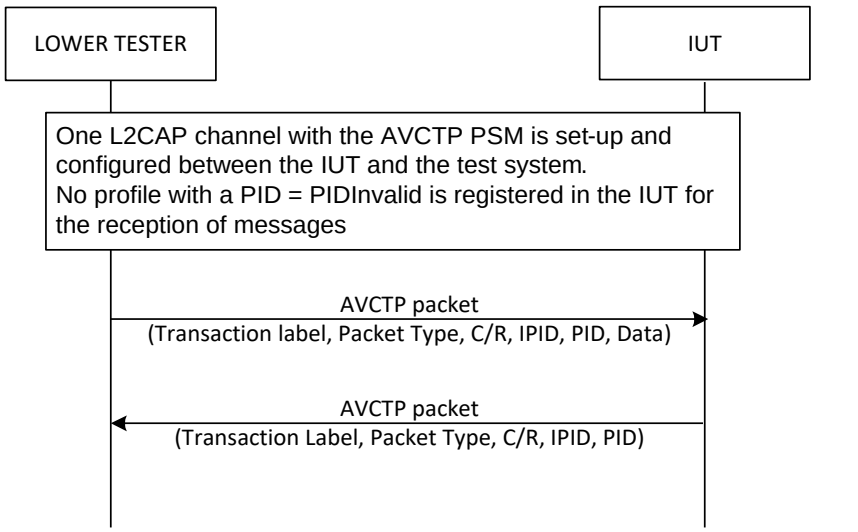

Figure 4.10: AVCTP/TG/NFR/BI-01-C [TG Handles Control Message with Invalid PID] – MSC

- Expected Outcome

  Pass verdict

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

  - Transaction label = $TRANS_{Test\_System}$
  - Packet_type = 00 (Non-fragmented Message)
  - C/R = 1 (Response Message)
  - IPID = 1 (Invalid Profile)
  - PID = $PID_{Invalid}$ (as assigned by the Lower Tester)

  No message information field is present.

4.4 Fragmented Messages

Verify the correct handling of fragmented messages on both CT and TG sides.

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

- Test Purpose

  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 $MTU_{LowerTester}$ bytes.
  - The PID identifying the profile using AVCTP is defined as PID_{Test_System}.
  - The BD_ADDR of the Lower Tester is defined as $BD\_ADDR_{Test\_System}$.

- Test Procedure

  1. If the Upper Tester and the 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 = 1
     - PID = PID_{Test_System}
     - Data = $ADDRESS_{data\_buffer}$ (Buffer holding $DATA[]_{Upper\_Tester}$)
     - Length = $L_{DATA[]Upper\_Tester}$ = $3*MTU_{LowerTester}$ - 6bytes

     If the Upper Tester or the IUT does 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 $MTU_{LowerTester}$ limit.

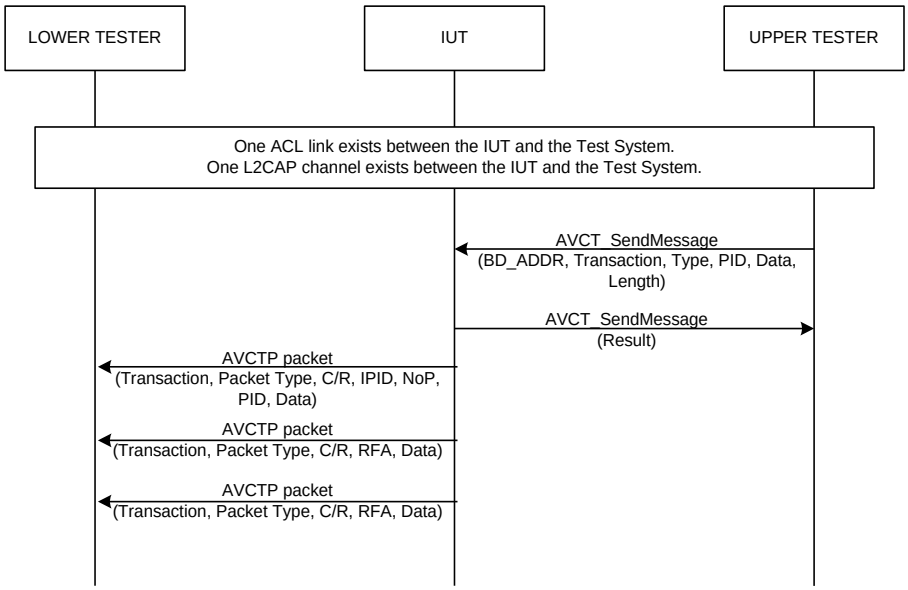

Figure 4.11: AVCTP/CT/FRA/BV-01-C [CT Fragments Command Messages] – MSC

- Expected Outcome

  Pass verdict

  If the Upper Tester and the 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 parameters:

  - 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 (length = $MTU_{LowerTester}$ -4)

  AVCTP continuation packet parameters:

  - Transaction = $TRANS_{Test\_System}$
  - Packet_type = 10 (Continue Packet)
  - C/R = 0 (Command Message)
  - RFA = 0
  - Data = Second part of the message information (length = $MTU_{LowerTester}$ -1)

  AVCTP end packet parameters:

  - Transaction = $TRANS_{Test\_System}$
  - Packet_type = 11 (End Packet)
  - C/R = 0 (Command Message)
  - RFA = 0
  - Data = Third part of the message information (length = $MTU_{LowerTester}$ -1)

AVCTP/TG/FRA/BV-02-C [TG Fragments Response Messages]

- Test Purpose

  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 the 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 $L_{DATA[\ ]LowerTester}$)

     The message has a data length of $L_{DATA[\ ]LowerTester} = MTU_{LowerTester}$ - 3bytes.

  2. If the Upper Tester and the 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* MTULowerTester -6 bytes

     If the Upper Tester or the IUT does 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* $MTU_{LowerTester}$ -6 bytes.

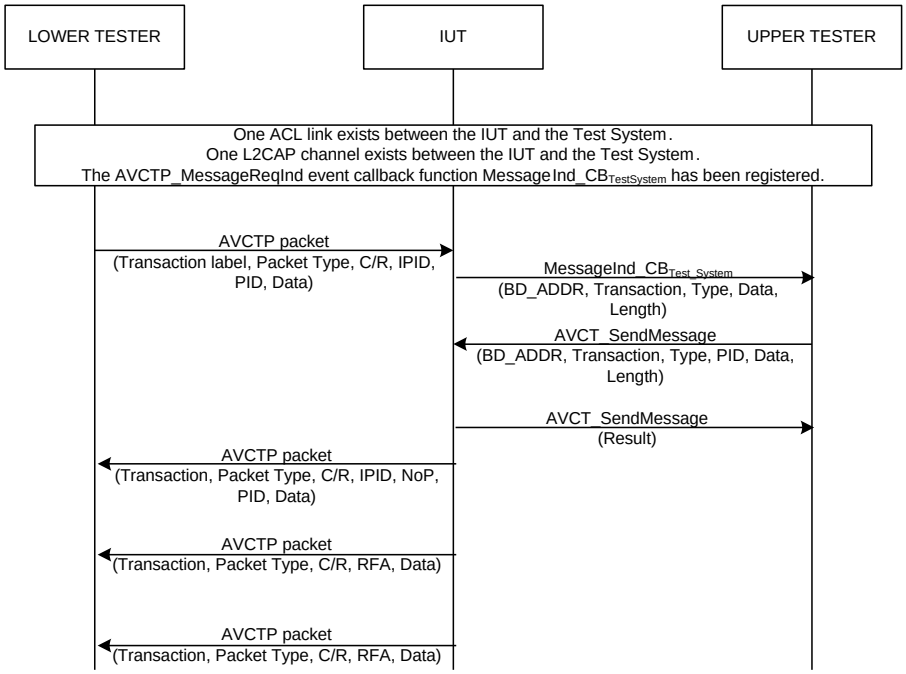

Figure 4.12: AVCTP/TG/FRA/BV-02-C [TG Fragments Response Messages] – MSC

- Expected Outcome

  Pass verdict

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

  - BD_ADDR = BD_ADDR$_{Test_System}$
  - Transaction = TRANS$_{Test_System}$
  - Type = 0x01 (Command Message)
  - Data = ADDRESS$_{data_buffer}$ (Buffer holding DATA[]$_{Lower\_Tester}$)
  - Length = L$_{DATA[\ ]LowerTester}$

  If the Upper Tester and the 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$_{Test_System}$
  - Packet_type = 01 (Start Packet)
  - C/R = 1 (Response Message)
  - IPID = 0 (Valid Profile)
  - NoP = 3
  - PID = PID$_{Test_System}$
  - Data = First part of the message information (length = MTU$_{LowerTester}$ -4)

AVCTP continuation packet:

- Transaction = $TRANS_{Test\_System}$
- Packet_type = 10 (Continue Packet)
- C/R = 1 (Response Message)
- RFA = 0
- Data = Second part of the message information (length = $MTU_{LowerTester}$ -1)

AVCTP end packet:

- Transaction = $TRANS_{Test\_System}$
- Packet_type = 11 (End Packet)
- C/R = 1 (Response Message)
- RFA = 0
- Data = Third part of the message information (length = $MTU_{LowerTester}$ -1)

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

- Test Purpose

  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 the 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 a 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 the 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.

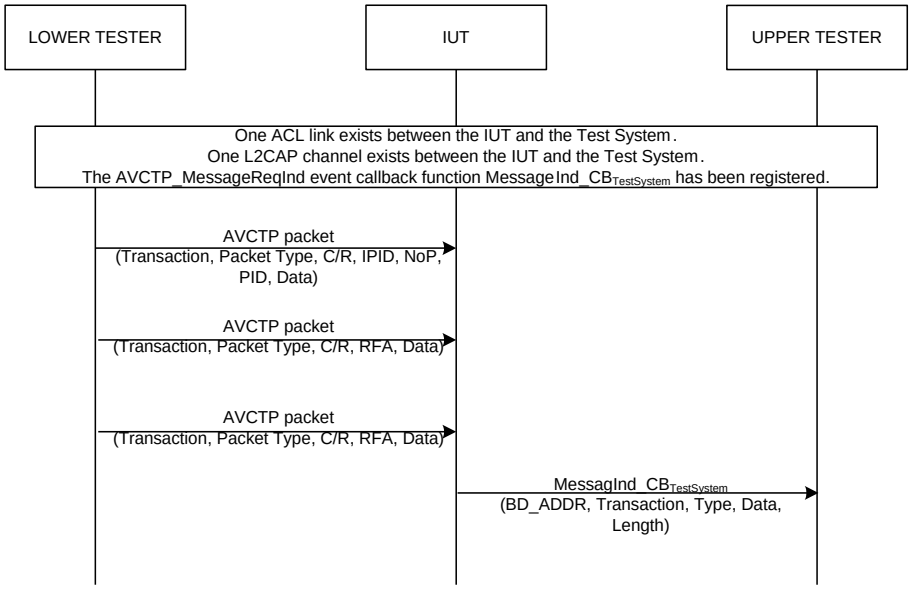

Figure 4.13: AVCTP/TG/FRA/BV-03-C [TG Reassembles Fragmented Command Message] – MSC

- Expected Outcome

  Pass verdict

  When the Upper Tester and the 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$_{TestSystem}$ callback function with the following parameters:

  - BD_ADDR = $BD\_ADDR_{Test\_System}$
  - Transaction = $TRANS_{Test\_System}$
  - Type = 0x01 (Command Message)
  - Data = $ADDRESS_{data\_buffer}$ (Buffer holding $DATA[]_{Lower\_Tester}$)
  - Length = $L_{DATA[]Lower\_Tester}$

  When the Upper Tester or the 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.

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

- Test Purpose

  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 the 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 the 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 = 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 the IUT does 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 three AVCTP packets with the following fields:
     AVCTP start packet:

     - $Transaction = TRANS_{Test\_System}$
     - 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:

     - $Transaction = TRANS_{Test\_System}$
     - Packet_type = 10 (Continue Packet)
     - C/R = 1 (Response message)
     - RFA = 0
     - Data = Second part of $DATA[]_{Lower\_Tester}$

AVCTP end packet:

- Transaction = $TRANS_{Test\_System}$
- Packet_type = 11 (End Packet)
- C/R = 1 (Response Message)
- RFA = 0
- Data = Third part of $DATA[]_{Lower\_Tester}$

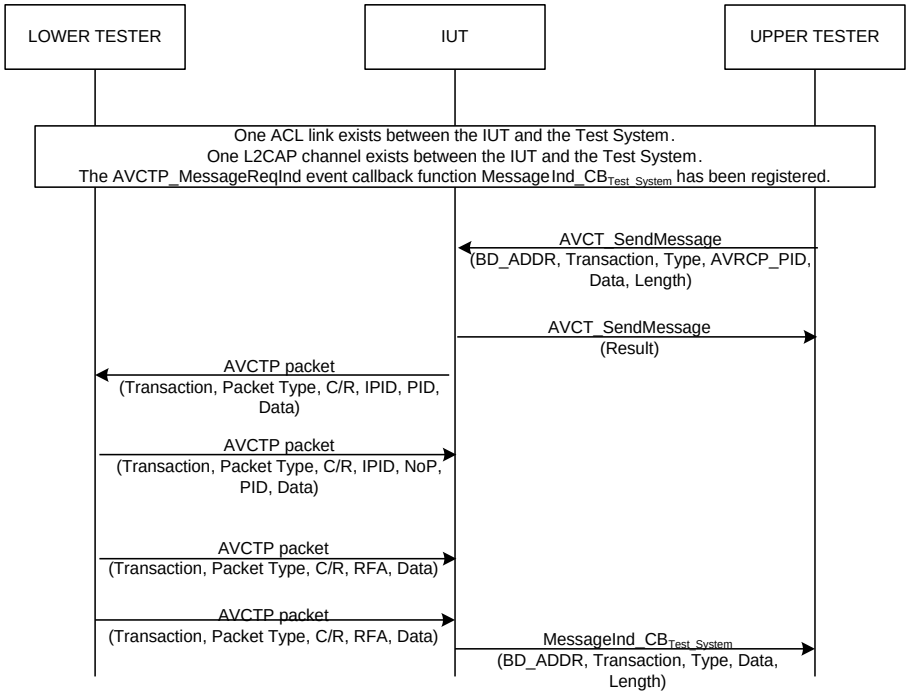

Figure 4.14: AVCTP/CT/FRA/BV-04-C [CT Reassembles a Fragmented Response Message] – MSC

- Expected Outcome

  Pass verdict

  When the Upper Tester and the 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 the IUT conform to the AVCTP Upper Interface for message event registration, the IUT calls the $MessageInd\_CB_{TestSystem}$ 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 IUT is 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 logical expression based on specific entries from the associated ICS document. Contains a logical expression (using the operators AND, OR, NOT as needed) based on specific entries from the applicable ICS document(s). The entries are in the form of y/x references, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS document for Audio/Video Control Transport Protocol (AVCTP) [4].

Feature: A brief, informal description of the feature being tested.

Test Case(s): The applicable test case identifiers are required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported. Further details about the function of the TCMT are elaborated in [6].

For the purpose and structure of the ICS/IXIT, refer to [6].

Item	Feature	Test Case(s)
AVCTP 2/11	Connect Request	AVCTP/CT/CCM/BV-01-C
AVCTP 3/11	Connect Request	AVCTP/TG/CCM/BV-01-C
AVCTP 2/12	Disconnect Request	AVCTP/CT/CCM/BV-02-C
AVCTP 3/12	Disconnect Request	AVCTP/TG/CCM/BV-02-C
AVCTP 2/9	Event registration for connection request	AVCTP/CT/CCM/BV-03-C
AVCTP 3/9	Event registration for connection request	AVCTP/TG/CCM/BV-03-C
AVCTP 2/10	Event registration for disconnection request	AVCTP/CT/CCM/BV-04-C
AVCTP 3/10	Event registration for disconnection request	AVCTP/TG/CCM/BV-04-C
AVCTP 2/14	Support for multiple AVCTP channel establishment requests	AVCTP/CT/CCM/BI-01-C
AVCTP 2/13	Send non-fragmented message (CT)	AVCTP/CT/NFR/BV-01-C
AVCTP 3/13	Send non-fragmented message (TG)	AVCTP/TG/NFR/BV-02-C
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	Event registration for message reception (TG)	AVCTP/TG/NFR/BV-03-C
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	Event registration for message reception (CT)	AVCTP/CT/NFR/BV-04-C

Item	Feature	Test Case(s)
AVCTP 3/2 OR AVCTP 3/3 OR AVCTP 3/4 OR AVCTP 3/5 OR AVCTP 3/6 OR AVCTP 3/7	Transaction label management (TG) Packet type field management (TG) Message type field management (TG) PID field management (TG) IPID field management (TG) Message information management (TG)	AVCTP/TG/NFR/BI-01-C
AVCTP 2/1 AND AVCTP 2/13	Send fragmented message (CT)	AVCTP/CT/FRA/BV-01-C
AVCTP 3/1 AND AVCTP 3/13	Send fragmented message (TG)	AVCTP/TG/FRA/BV-02-C
AVCTP 3/1	Fragmented message reassembly (TG)	AVCTP/TG/FRA/BV-03-C
AVCTP 2/1	Fragmented message reassembly (CT)	AVCTP/CT/FRA/BV-04-C

Table 5.1: Test case mapping

6 Revision history and acknowledgments

Revision History

Publication Number	Revision Number	Date	Comments
0	Version 1.0	2003-05-01	Updated title and header
1	Version 1.2	2004-04-14	Updated Disclaimer and Copyright Notice. Clerical changes.
2	1.1.1r1	2004-12-29	Editorial and format change. Incorporate TSE 700 for TP/CCM/BI-01-C Incorporate TSE 702 for TCMT.
3	1.2.1	2005-02-10	Document number change. Incorporate TSE 703 for TP/NFR/BV-03-C. Incorporate TSE 708 for TCMT. Incorporate TSE 709 for TCMT. Incorporate TSE 710 for TP/FRA/BV-03-C. Incorporate TSE 711 for TP/FRA/BV-04-C. Incorporate TSE 712 for TP/NFR/BI-01-C.
4	1.2.2r1-02 1.2.2r2 1.2.2	2005-08-15 – 2005-11-03	Incorporate TSE 783 for TP/FRA/BV-01-C, TP/FRA/BV-03-C, TP/FRA/BV-04-C Incorporate TSE 741 for TP/NFR/BV-01-C and TP/NFR/BV-02-C Corrections to TP/CCM/BI-01-C in response to AV WG input Corrections to Test Procedure for TP/NFR/BV-03-C, Corrected cross refs. for TP/NFR/BV-04-C/TP/FRA/BV-02-C Prepare for publication
	1.2.3r0	2006-11-01	Add Conformance section TSE 1814: TP/CCM/BV-03-C TSE 1865: Fix graphic for TP/CCM/BV-04-C
	1.2.3r1	2006-11-30	Accept changes Input reviewer's comments Remove table of abbreviations, list of tables, and list of figures Move Section 5, References, to Section 2.2 Move MSCs to their respective test cases Merge table cells for selection criteria that pertain to a single test case
5	1.2.3	2006-01-08	Prepare for publication.
	1.2.4r0	2008-02-01	TSE 2391: TP/FRA/BV-01-C,TP/FRA/BV-02-C
6	1.2.4	2008-04-01	Prepare for publication.
	1.2.5r0	2008-10-01	TSE 2593 TP/NFR/BI-01-C, MSC correction
7	1.2.5	2008-11-25	Prepare for publication.

Publication Number	Revision Number	Date	Comments
	1.3.6r0-r2	2010-12-17 - 2011-03-01	TSE 3820 TP/CCM/BV-01-C, TP/CCM/BV-02-C, TP/CCM/BV-03-C, TP/CCM/BV-04-C Accepted previous changes. Changed version number in previous row. Input reviewer's comments for TSE 3820: 1) update ToC, 2) Update Pass verdict for TP/CCM/BV-01-C, TP/CCM/BV-02-C, TP/CCM/BV-04-C per TSE 2nd .doc attachment. Input Alicia C's comments for TP/CCM/BV-01-C, TP/CCM/BV-02-C per TSE 3820
8	1.3.6	2011-07-21	Prepare for publication.
	1.3.7r0	2011-09-13	TSE 4409: TCMT for TP/NFR/BV-01-C, TP/NFR/BV-02-C
	1.3.7r1	2011-10-25	Correction to TP/NFR/BV-01-C in TCMT by AC
9	1.3.7	2012-03-30	Prepare for publication.
	1.3.8r0	2012-05-18	TSE 4607: TP/NFR/BI-01-C: update MSC to remove "Data" from response.
	1.4.0r0	2012-05-31	Updated specification revision labeling. Application of BTI recommendations for test case bulleting, indenting and fail verdict text. Corrected hypertext font coloring.
	1.4.0r1	2012-06-06	Updated Section 2 to comply with latest TS template.
	1.4.0r2	2012-06-19	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.
10	1.4.0	2012-07-24	Prepare for publication.
	1.4.1r01	2017-01-27	Converted TS template.
	1.4.1r02	2017-04-05	Converted to new Test Case ID conventions as defined in TSTO v4.1
11	1.4.1	2017-07-03	Approved by BTI. Prepared for TCRL 2017-1 publication.
	p12r00	2024-01-04 – 2024-01-05	TSE 24739 (rating 1): Updated the document to align with the latest conventions, including converting the publication numbering scheme and updating the copyright page.
12	p12	2024-07-01	Approved by BTI on 2024-05-22. Prepared for TCRL 2024-1 publication.

Acknowledgments

Name	Company
David Trainor	CSR
Morgan Lindqvist	Ericsson
Fisseha Mekuria	Ericsson
Michinori Masuda	Matsushita Electric Industrial
Tsuyoshi Okada	Matsushita Electric Industrial
Christian Bouffioux	Philips
Henk Koopmans	Philips
Emmanuel Mellery	Philips
Piotr Polak	Philips
Harumi Kawamura	Sony
Rudiger Mosig	Sony
Hiroyasu Noguchi	Sony
Yoshinari Kumaki	Toshiba
Ichiro Tomoda	Toshiba