

Link Layer (LL)

Bluetooth® Test Suite

- **Revision:** LL.TS.5.1.0
- **Revision Date:** 2018-12-07
- **Group Prepared By:** BTI
- **Feedback Email:** bti-main@bluetooth.org

Abstract:

This document defines test structures and procedures for the conformance test of Bluetooth products implementing the low energy Link Layer Protocol.



Revision History

Revision History	Date	Comments
D09r04	2009-05-06	Revision 09 approved by BTI (include review comments resolution).
D09r05	2009-06-09	Editorial review and markup.
D09r06	2009-10-30	*MSCs added. *Test Procedures clarified. *Updated to LE Draft v1.0 *TP/SEC/SLA/BV-04-C and TP/SEC/MAS/BV-03-C added.
D09r08	2009-11-18	All changes accepted as agreed in LE Test group F2F the 18th
D09r07	2009-11-18	* Added comments from Magnus Sommansson. * Added comments from Paul Vanoostende. * Typo errors updated. * Errata 3127, 3221 incorporated.
LL.TS/Tokyo.1.0.0 r1-15	2009-11-24- 2009-12-15	Updated references Editorial adjustments to harmonize with other Tokyo TS documents Incorporate errata 3299; affects 2 test cases: TP/SEC/SLA/BV-04C and TP/SEC/MAS/BV-03-C Incorporate 3316 affects TP/CON/MAS/BV-08,09-C, TC/CON/SLA/BV-11,12-C: Modifications in MSCs and procedures., AND TP/CON/SLA/BI-05-C and TP/CON/MAS/BI-04-C, modified due to the inclusion of the T_Terminate timer in the termination procedure Modification of test cases, using updated control procedure Added TP/CON/SLA/BI-02-C and TP/CON/MAS/BI-02-C, to test the behavior of the IUT when it uses the T_Terminate timer TP/CON/SLA/BV-17-C:MSC corrected TP/CON/MAS/BV-16-C:MSC corrected TP/CON/SLA/BI-04-C: Test case mapping corrected TP/CON/MAS/BI-02-C: MSC name corrected TP/CON/SLA/BV-10-C: Typo error in MSC corrected TIM/SLA group, SEC/SLA/BI group, TP/SEC/MAS/BI-05, 06: MSCs and procedures corrected Preamble MSC corrected TP/CON/INI/BI-02-C: 'reason' parameter value corrected SEC group test cases updated TP/SEC/SLA/BI-03-C and TP/SEC/SLA/BI-04-C: Verdict section corrected Test Case Mapping corrected TP/SEC/MAS/BI-03-C: MSC corrected Test procedure of TIM group test cases corrected



Revision History	Date	Comments
		<p>Change to final document number structure TPG integration issues with TCMTs corrected for TP/DDI/SCN/BV-12-C; TP/CON/SLA/BV-21-C; TP/CON/MAS/BV-16-C & BV-17-C & BV-18-C & BV-22-C,; TP/ENC/SCN/BI-02-C Corrected 8/2 to 7/2 in TP/SEC/MAS/BI-05-C; TP/SEC/MAS/BI-06-C</p>
LL.TS/4.0.0	2009-12-17	Prepare for publication
LL.TS/4.0.1r0	2010-06-07	<p>NOTE: In descending order:</p> <p>TSE 3563: TP/DDI/ADV/BV-11-C: Updated initial condition and test procedure</p> <p>TSE 3559: TP/DDI/SCN/BV-05-C: Removed data fields from ADV_DIRECT_IND</p> <p>TSE 3557: TP/SEC/SLA/BV-02-C: Update MSC and test procedure</p> <p>TSE 3549: TP/SEC/SCN/BV-01-C: Update MSC and test procedure</p> <p>TSE 3542: TP/DDI/ADV/BV-09-C: Update test procedure steps 7&9</p> <p>TSE 3532: TP/TIM/SLA/BV-04-C: Remove Note.</p> <p>TSE 3524: TP/CON/SLA/BI-05-C, TP/CON/MAS/BI-04-C: add alternative MSCs</p> <p>TSE 3519: TP/SEC/MAS/BI-01-C: Correct MSC</p> <p>TSE 3518: TP/SEC/MAS/BV-01-C, TP/SEC/MAS/BV-03-C, TP/SEC/MAS/BI-01-C, TP/SEC/MAS/BI-03-C, TP/SEC/MAS/BI-04-C: correct MSCs</p> <p>TSE 3494: TP/SEC/SLA/BI-01-C: Update MSC and test procedure</p> <p>TSE 3493: TP/DDI/ADV/BV-09-C: Correct test procedure step 8</p> <p>TSE 3492: TP/CON/INI/BV-06-C, TP/CON/INI/BV-07-C: Correct test procedure.</p> <p>TSE 3486: TP/DDI/SCN/BV-05-C: Add step to test procedure.</p> <p>TSE 3483: TP/CON/SLA/BV-05-C, TP/CON/SLA/BV-06-C, TP/CON/MAS/BV-04-C, TP/CON/MAS/BV-05-C: Correct MSCs and test procedures pass verdicts.</p> <p>TSE 3472: TP/CON/INI/BI-02-C: Update purpose, MSC, test proc, pass verdict</p> <p>TSE 3471: TP/CON/INI/BI-02-C: update MSC, test proc</p> <p>TSE 3470: TP/ENC/INI/BI-01-C: Update purpose, MSC, pass verdict</p> <p>TSE 3467: TP/CON/SLA/BV-04-C, TP/CON/MAS/BV-03-C: Update MSC and test procedures.</p> <p>TSE 3460: TP/CON/SLA/BV-05-C: Modify MSC?</p> <p>TSE 3410: TP/CON/INI/BV-01-C: Modify Reference, test procedure, and Pass verdict</p>



Revision History	Date	Comments
		<p>NOTE: In ascending order: (in case of superseding TSEs)</p> <p>TSE 3371: TP/CON/SLA/BV-04-C: Update TCMT</p> <p>TSE 3374: TP/CON/SLA/BV-16-C, TP/CON/SLA/BV-17-C: Update MSCs and test procedures</p> <p>TSE 3375: TP/CON/MAS/BV-02-C: Update MSC latency value</p> <p>TSE 3376: TP/SEC/SLA/BV-02-C, TP/SEC/SLA/BV-03-C: Update MSCs and test procedures</p> <p>TSE 3377: TP/SEC/SLA/BV-04-C: Update Verdicts</p> <p>TSE 3378: TP/DDI/ADV/BV-11-C: Correct MSC</p> <p>TSE 3379: TP/CON/ADV/BI-01-C: Test purpose, caption</p> <p>TSE 3380: TP/CON/SLA/BV-02-C: Update verdicts</p> <p>TSE 3388: TP/FRH/ADV/BV-01-C: Correct TCMT</p> <p>TSE 3392: TP/PAC/MAS/BV-01-C: Correct TCMT</p> <p>TSE 3393: TP/CON/MAS/BV-03-C: Correct TCMT</p> <p>TSE 3394: TP/CON/SLA/BV-05-C: Correct TCMT</p> <p>TSE 3395: TP/CON/SLA/BV-06-C: Correct TCMT</p> <p>TSE 3399: TP/SEC/ADV/BV-01-C: Correct test purpose</p> <p>TSE 3400: TP/DDI/ADV/BV-15-C: Correct MSC</p> <p>TSE 3401: TP/DDI/SCN/BV-01-C, TP/DDI/SCN/BV-10-C, TP/DDI/SCN/BV-11-C, TP/DDI/SCN/BV-12-C: Modify test procedure</p> <p>TSE 3403: TP/SEC/MAS/BV-02-C: Correct Fail verdict</p> <p>TSE 3404: TP/SEC/MAS/BI-06-C: Correct step 4, test procedure</p> <p>TSE 3405: TP/TIM/MAS/BV-01-C, TP/TIM/MAS/BV-02-C: Correct jitter values in test procedures</p>
4.0.1r1	2010-06-16	Accepted changes. Entered Antonio's review comments for SEC/SLA/BV-02, CON/ADV/BI-01, SEC/MAS/BV-02, CON/INI/BV-01, ENC/INI/BI-01, CON/INI/BI-02, CON/SLA/BV-05, CON/MAS/BV-04, CON/MAS/BV-05, DDI/SCN/BV-05, DDI/SCN/BV-05, SEC/MAS/BV-01, SEC/MAS/BV-03, SEC/SCN/BV-01.
4.0.1r2	2010-06-17	Fix caption for CON/ADV/BI-01 Fixed wording in CON/SLA/BV-05 Fixed MSC for SEC/MAS/BV-01
4.0.1	2010-06-24	Prepare for publication
4.0.2r0-r3	2010-11-24 - 2011-02-07	<p>TSE 3517: TP/SEC/MAS/BI-07-C: update purpose; new test cases TP/SEC/MAS/BV-04-C, TP/SEC/MAS/BV-05-TP/SEC/MAS/BI-08-C, TP/SEC/MAS/BI-09-C</p> <p>TSE 3634 (supersedes TSE 3641):</p> <ul style="list-style-type: none"> -updates to TP/TIM/SLA/BV-02-C, TP/TIM/SLA/BV-04-C, TP/TIM/BV-05-C, -: new test cases TP/TIM/ADV/BV-01-C, TP/TIM/ADV/BV-02-C, TP/TIM/ADV/BV-06-C, TP/TIM/ADV/BV-07-C,



Revision History	Date	Comments
		<p>TSE 3761: TP/CON/SLA/BV-02-C, TP/CON/MAS/BV-02, TP/CON/MAS/BV-22-C; updated MSCs</p> <p>TSE 3927: TP/DDI/ADV/BV-15-18, TP/DDI/SCN/BV-05, TP/DDI/SCN/BV-12-C</p> <p>TSE 3939: TP/CON/SLA/BV-21-C</p> <p>TSE 4098: TP/DDI/ADV/BV-01-C, TP/DDI/ADV/BV-02-C, TP/DDI/ADV/BV-15-C: edit Pass verdict.</p> <p>TSE 4111: TP/DDI/ADV/BV-06-C, TP/DDI/ADV/BV-07-C TP/DDI/ADV/BV-09-C, TP/DDI/ADV/BV-11-C, TP/CON/ADV/BV-04-C: edit Pass/Fail verdicts</p> <p>Input corrections per AT4Wireless review for TSE 3517</p> <p>TP/SEC/MAS/BI-07 verdicts</p> <p>TPSEC/MAS/BI-09 MSC</p> <p>TP/SEC/MAS/BV-04 test procedure</p> <p>TP/SEC/MAS/BV-05 pagination</p> <p>TCMT:-consolidate rows</p> <p>TSE 3634</p> <p>Rev History for TSE 3641</p> <p>TP/TIM/ADV/BV-01: MSC caption</p> <p>TP/TIM/ADV/BV-02: MSC caption</p> <p>TP/TIM/SLA/BV-04: Init Condition SCA value</p> <p>TP/TIM/SLA/BV-05: Init Condition SCA value, MSC</p> <p>TP/TIM/SLA/BV-06: MSC caption</p> <p>TP/TIM/SLA/BV-07: MSC caption (did not see Notes formatting issue)</p> <p>TSE 3297: Note that this is 3927 and is correct as is.</p> <p>TSE 4098: TP/DDI/ADV/BV-02; update verdicts (remove 90%)</p> <p>TSE 4111:</p> <p>TP/DDI/ADV/BV-07, update MSC</p> <p>TPDDI/ADB/BV-09; update MSC</p> <p>TP/DDI/ADV/BV-11: update MSC</p> <p>TP/CON/ADV/BV-04-update MSC, correct test proc, step 4.</p> <p>TSE 3634: TP/TIM/SLA/BV-05: Changed Initial Condition value from -20ppm to +20ppm</p> <p>TSE 3927: TP/DDI/ADV-18: Fixed packet label in MSC</p> <p>TP/CON/ADV/BV-04- Fixed graphic to fit on one page</p> <p>Change reference from [2] to [3] in test case Reference sections.</p>
4.0.2	2011-07-18	Prepare for publication.
4.0.3r0	2011-11-08	<p>TSE 3666 TP/CON/SLA/BI-05-C for Errata 3550. Add paragraph.</p> <p>TSE 3886: TP/CON/ADV/BI-01-C: Pass verdict, Step 9 of Test. Procedure.</p> <p>TSE 3887: TP/CON/INI/BI-02-C: Pass verdict, Step 9 of Test. Procedure.</p>



Revision History	Date	Comments
		<p>TSE 4097: TP/DDI/ADV/BV-01-C: Fix formula.</p> <p>TSE 4349: TP/CON/MAS/BV-19-C, TP/CON/MAS/BI-04-C, TP/SEC/SLA/BI-01-C, TP/SEC/MAS/BI-03-C, TP/SEC/MAS/BI-04-C, TP/SEC/MAS/BI-07-C, TP/SEC/MAS/BI-08-C, TP/SEC/MAS/BI-09-C, update Test procedures steps and MSCs, Pass and Fail verdicts.</p> <p>TSE 4351: new test case TP/SEC/SLA/BV-05-C, update TCMT</p> <p>TSE 4380: TP/SEC/MAS/BI-07-C, TP/SEC/MAS/BI-09: update Pass verdicts-C</p> <p>TSE 4459: TP/CON/MAS/BV-02-C: Change Test Procedure, Pass verdict.</p>
4.0.3r1	2011-11-09	<p>TSE 3666: TP/CON/SLA/BI-05-C per Antonio's instructions.</p> <p>TSE 4349: TP/CON/MAS/BV-19-C. Corrected placement of Optional box.</p>
4.0.4r0	2012-05-18	TSE 4631: TP/DDI/ADV/BV-06-C, TP/DDI/ADV/BV-07-C, TP/DDI/ADV/BV-09-C: MSC corrections (HCl_*LE*_....)
4.0.4r1	2012-07-05	<p>TSE 4814: TP/CON/ADV/BI-01-C MSC and verdict updates</p> <p>TSE 4816: TP/CON/INI/BI-02-C MSC and verdict updates</p>
4.0.4	2012-07-24	Prepare for publication.
4.0.5r1	2013-05-31	<p>TSE 4975: Updated test procedure of TP/SEC/MAS/BI-08-C to change to the correct error code.</p> <p>TSE 5082: Updated test procedure and MSC for TP/DDI/ADV/BV-11-C.</p> <p>TSE 5143: Updated test procedure and MSC for TP/SEC/MAS/BV-03-C.</p> <p>TSE 5144: Updated test case description, MSC, and Step 5 of the test procedure for TP/SEC/MAS/BI-08-C.</p>
4.0.5r2	2013-06-03	BTI Review, Comments from Miles.
4.0.5	2013-07-02	Prepare for Publication
4.0.6rT, 4.0.6rTr3, 4.0.4rTr4	2013-07-07, 2013-09-25	<p>Template Conversion:</p> <ul style="list-style-type: none"> - Fail Verdicts Removed - New Pass/Fail Verdict Criteria section added - Removal of sections marked "N/A"
4.1.0r01	2013-09-25	Low Duty Cycle Directed Advertising CR
4.1.0r02	2013-10-10	<p>LE Ping CR</p> <p>TSE 5388: In Figure 4.1 updated the wording to read "the BER on the RF is less than 0.1%".</p>
4.1.0r03	2013-10-14	LE Link Layer Topology CR
4.1.0	2013-12-03	Prepare for Publication
4.1.1r00	2014-10-21	TSE 5665: Updating MSC and step 7 and 8 in the Test Procedure for TP/CON/INI/BI-02-C.



Revision History	Date	Comments
		<p>TSE 5785: Updated Step 1 and Step 6 to read “HCI_Command_Complete” instead of “HCI_Command_Status” in TP/FRH/MAS/BV-01-C.</p> <p>TSE 5981: Updated MSC for TP/CON/SLA/BI-05-C.</p> <p>TSE 5528: Addition of three new test cases and associated TCMT entries: TP/SEC/SLA/BI-05-C, TP/SEC/MAS/BV-12-C and TP/SEC/MAS/BV-13-C. Updated TCMT mapping for TP/SEC/SLA/BI-03-C, TP/SEC/SLA/BI-04-C.</p>
4.1.1r01	2014-11-05	BTI Review, Dave, capitalized “Upper Tester” in new test cases, TP/SEC/MAS/BV-12-C and TP/SEC/MAS/BV-13-C.
4.2.0r00	2014-11-17	Integrated changes from Section 6 of Core_LE_Data_Length_Extensions_TEST.CRR01 and Sections 1.7 & 1.9 of Core_Enhanced_Privacy_1_2.TS.CR.R05
4.2.0r01	2014-11-20	Integrated Reviews from Mayank and Rasmus
4.2.0r02	2014-11-24	Added captions to new MSCs. Minor editorial fixes
4.2.0r03	2014-11-25	Addressed outstanding comments from Mayank in 4.3.4 & 4.3.5.
4.2.0r04	2014-11-25	Review by Dave. Minor editorial changes
4.2.0	2014-12-04	Prepare for TCRL 2014-2 publication
4.2.1r00	2015-05-06	<p>TSE 6122: Added new test case TP/SEC/MAS/BV-14-C for unexpected Data Channel PDUs during encryption start (IUT Master). Updated TCMT accordingly.</p> <p>TSE 6181: Updated TP/DDI/SCN/BV-14-C to change random address to resolvable private address.</p> <p>TSR 6174: Updated TP/DDI/SCN/BV-14-C to correct Scan Filter Policy value from 0x04 to 0x02</p> <p>TSE 6292: Updated MSC for TP/DDI/SCN/BV-15-C</p> <p>TSE 6033: Editorial correction to test procedure for TP/CON/SLA/BV-10-C</p> <p>TSE 6034: Updated MSC and step 12 of test procedure for TP/DDI/ADV/BV-19-C</p> <p>TSE 6084: Changed Data length from FF to FFFFFFFF in TP/CON/MAS/BV-02-C & TP/CON/MAS/BV-22-C</p> <p>TSE 6248: Corrected TCMT mapping for TP/SEC/SCN/BV-01-C</p> <p>TSE 6134: Replaced MSCs for TP/CON/INI/BV-01-C, TP/CON/INI/BV-02-C, TP/CON/INI/BV-04-C, TP/CON/INI/BI-02-C, TP/ENC/INI/BI-01-C to allow jitter in the first packet of a connection</p> <p>TSE 6154: Updated MSC and test procedure in TP/SEC/SLA/BI-05-C to allow option to send LL_ENC_RSP.</p> <p>TSE 6260: Updated item mapping in TCMT to remove redundant mapping to Table 1 items.</p> <p>TSE 6213: Clarified tests by replacing “empty data packet” and similar terms with “Correctly formatted LL Data Channel PDU” in</p>



Revision History	Date	Comments
		text and MSCs. Test groups affected: TP/CON/ADV & TP/CON/INI
4.2.1r01	2015-05-18	Following review on BTI call May 18 2015, extended changes from TSE 6213 through the entire CON/ADV & CON/INI test groups.
4.2.1r02	2015-06-15	TSE 6149: For tests TP/CON/SLA/BV-19-C, TP/CON/SLA/BI-05-C, and TP/CON/MAS/BV-20-C – updated MSCs and test procedures to allow for cases where the LL autonomously initiates the LL_VERSION_IND transaction.
4.2.1	2015-07-14	Editorial updates to correct specification references Prepared for publication
4.2.2r00	2015-10-13	<p>TSE 6491 & 6655: Corrected terminology in MSCs from HCI_LE_Set_Random_Private_Address_Timeout to HCI_LE_Set_Resolvable_Private_Address_Timeout. Affects TP/CON/INI/BV-08-C, TP/CON/INI/BV-09-C, TP/CON/INI/BV-10-C, TP/CON/INI/BV-11-C, TP/CON/INI/BV-12-C, TP/DDI/SCN/BV-13-C, TP/DDI/SCN/BV-14-C, TP/DDI/SCN/BV-15-C, TP/DDI/SCN/BV-16-C, TP/DDI/SCN/BV-17-C, TP/DDI/SCN/BV-18-C, TP/SEC/ADV/BV-03-C, TP/SEC/ADV/BV-05-C, TP/SEC/ADV/BV-07-C, TP/SEC/ADV/BV-08-C, TP/SEC/ADV/BV-10-C, TP/SEC/ADV/BV-11-C, TP/SEC/ADV/BV-12-C, TP/SEC/ADV/BV-14-C.</p> <p>TSE 6657: Updated MSC in TP/CON/INI/BV-09-C to account for test steps 11-15.</p> <p>TSE 6658: Deleted duplicate command in TP/SEC/ADV/BV-05-C.</p> <p>TSE 6659: Updated MSC in TP/SEC/ADV/BV-06-C to account for test step 7.</p> <p>TSE 6660: Corrected MSC in TP/SEC/ADV/BV-11-C and TP/SEC/ADV/BV-12-C by deleting procedure repetition.</p> <p>TSE 6661: Updated MSC in TP/SEC/ADV/BV-13-C to account for test steps 8-10.</p> <p>TSE 6708: Removed “It is verified Hop Length” from MSC in TP/FRH/ADV/BV-01-C</p> <p>TSE 6679: Changed LL_LENGTH_IND to LL_LENGTH_REQ in TP/CON/MAS/BV-36-C, TP/CON/MAS/BV-37-C, TP/CON/MAS/BV-38-C, TP/CON/MAS/BV-39-C, TP/CON/SLA/BV-35-C, TP/CON/SLA/BV-37-C, TP/CON/SLA/BV-39-C.</p> <p>TSE 6609: Added HCI Command Status response event to step 3 of TP/SEC/MAS/BV-14-C and updated MSC accordingly.</p> <p>TSE 6588: Updated TP/SEC/MAS/BV-11-C and TP/SEC/SLA/BV-11-C to specify Reject Op code. Updated TP/SEC/SLA/BV-04-C to clarify Lower Tester support conditions.</p> <p>TSE 6560: Edited Step 5 and Pass verdict of TP/DDI/SCN/BV-14-C to remove IUT requirement to resolve InitA address.</p>



Revision History	Date	Comments
		<p>TSE 6554: Corrected test procedure and Pass verdict in TP/SEC/MAS/BI-07-C to match previously updated MSC.</p> <p>TSE 6526: Reworded step 9 requirement from “needs to be” to “should be” and updated Pass verdict accordingly in TP/CON/INI/BV-11-C and 12-C.</p> <p>TSE 6365: Changed HCI_LE_Connection_Complete_Event to HCI_LE_Enhanced_Connection_Complete_Event in TP/SEC/ADV/BV-06-C MSC.</p> <p>TSE 6150: Updated MSC, test procedure, and verdict of TP/SEC/MAS/BV-12-C to add an alternative path through test to allow for the case where IUT has initiated LL_VERSION_IND.</p>
4.2.2	2015-12-22	Prepared for TCRL 2015-2 publication.
4.2.3r00	2016-02-12	TSE 6737: Unnecessary step deleted from test procedure of test case TP/SEC/ADV/BV-03-C.
4.2.3r01	2016-03-02	TSE 6865: Global edit. Changed ADV_DISCOVER_IND to ADV_SCAN_IND in body text and MSC (Figure 4.33).
4.2.3r02	2016-03-10	TSE 6652: Updated test case TP/ENC/SLA/BI-01-C: State, Test Procedure, Pass Verdict, and Notes.
4.2.3r03	2016-04-06	TSE 6103: Updated MSCs and test procedures for test cases TP/CON/SLA/BV-22-C, TP/CON/SLA/BV-23-C, TP/CON/MAS/BV-13-C, and TP/SEC/MAS/BV-11-C. Deleted entire section (4.3.4.39) for test case TP/CON/SLA/BI-06-C. Deleted row from TCMT for test case TP/CON/SLA/BI-06-C. Deleted 6/14 from TCMT for test case TP/CON/SLA/BI-05-C.
4.2.3r04	2016-04-12	<p>TSE 6623: Deleted “but which is generated using the IRK distributed to the IUT” from Test Procedure step 3 for test case TP/DDI/SCN/BV-14-C.</p> <p>TSE 6656: Updated MSC per BTI discussion and TSE comment #22926. Deleted steps 9–12 in the Test Procedure and the first sentence of the Pass Verdict for test case TP/CON/INI/BV-08-C.</p> <p>TSE 6941: Updated text for Section 4.1.5.7, Common Test Procedure Steps. Extensive editorial and technical updates to Section 4.1.5.7.3, Optional Test Steps.</p> <p>TSE 6970: Updated MSC and figure caption. Updated step 2 of the Test Procedure and the Pass Verdict for test cases TP/SEC/SLA/BV-07-C and TP/SEC/MAS/BV-07-C.</p> <p>TSE 6977: Updated first paragraph and Step 15 of Test Procedure for test case TP/DDI/ADV/BV-03-C.</p> <p>TSE 6983: Test case TP/SEC/ADV/BV-01-C updated: Section heading, test case description, Reference, Initial Condition State, Test Procedure (including MSC), and Pass Verdict. Updated TCMT, Feature name.</p> <p>TSE 6992: Optional HCI event added to MSC for test case TP/SEC/MAS/BV-14-C.</p> <p>TSE 6993: Typo corrected. “Step 6” changed to “Step 4.”</p>



Revision History	Date	Comments
		TSE 7015: Deleted second sentence of Pass Verdict of test case TP/DDI/ADV/BV-19-C. TSE 7016: Updated MSC and Pass Verdict for test cases TP/CON/SLA/BV-39-C and TP/CON/MAS/BV-40-C.
4.2.3r05	2016-04-25	Updated MSCs for test cases TP/CON/SLA/BV-23-C, TP/CON/MAS/BV-13-C, and TP/SEC/MAS/BV-11-C: "...FEATURE_REQ" and "...RSP" changed from solid to dashed arrows.
4.2.3	2016-07-13	Prepared for TCRL 2016-1 publication
5.0.0r00	2016-07-07	Integrated changes for Core Specification 5.0 release
5.0.0r01	2016-06-06	Test Spec Issue 7186: Updates test cases TP/CON/SLA/BV-45-C, TP/CON/SLA/BV-51-C, TP/CON/SLA/BI-09-C, and TP/CON/MAS/BV-46-C.
5.0.0r02	2016-06-28	Issue 7302: Deleted entire section and TCMT entries for test cases TP/DDI/SCN/BV-22-C and TP/DDI/SCN/BV-26-C. Deleted "autonomous" from "autonomous periodic advertising" for test cases TP/DDI/ADV/BV-26-C, TP/DDI/ADV/BV-33-C, TP/DDI/SCN/BV-21-C, and TP/DDI/SCN/BV-25-C. MSCs updated for TP/DDI/ADV/BV-26-C, TP/DDI/ADV/BV-33-C: "(Autonomous)" changed to "(Enable)." Issue 7280: Updated Advertising_Event_Properties values in MSC and Step 2 of test cases TP/CON/ADV/BV-06-C, TP/DDI/ADV/BV-21-C, and TP/DDI/ADV/BV-22-C.
5.0.0r03	2016-07-29	Issue 7279: Updated 25 test cases (TP/DDI/ADV/BV-21-C – 34-C*, TP/DDI/SCN/BV-19-C – 26-C**, TP/CON/ADV/BV-05-C – 06-C, and TP/CON/INI/BV-13-C): Updated section title, Figure, Test Procedure text and step numbering, Table, and Pass Verdict. *TP/DDI/ADV/BV-23-C: Deleted entire test case and TCMT entry. **TP/DDI/SCN/BV-22-C and 26-C: Changes not applicable. Entire test case deleted by Issue 7302. Issue 7399: For test case TP/DDI/ADV/BV-32-C, all binary values in table converted to hexadecimal.
5.0.0r04	2016-08-10	Issue 7487: Deleted duplicate test cases TP/FRH/SLA/BV-04-C and TP/FRH/SLA/BV-05-C.
5.0.0r05	2016-08-12	Issue 7465: Updated MSC, Pass Verdict, and Test Procedure (step 3) for test case TP/CON/SLA/BV-44-C.
5.0.0r06	2016-09-07	Issue 7532: TCMT updated: Moved references to test case TP/DDI/ADV/BV-24-C and TP/DDI/ADV/BV-28-C. Deleted row* referencing Feature "Extended Advertising, Secondary Channels" (*Note: Edit to already modified text. Change could not be tracked). Item "LL 3/13" changed to "LL 3/12". Deleted "AND LL 3/13" and "Secondary Channels". Issue 7533: TCMT updated for reference to test case TP/CON/INI/BV-14-C. Changed Item "LL 5/4 AND LL 1/3" to "LL 4/7 AND LL 5/1".



Revision History	Date	Comments
		<p>Issue 7571: Added new row and mapping for LL 3/11 to TCMT. Reference to test case "TP/DDI/ADV/BV-33-C" moved down from adjacent row above. Changed mapping for LL "3/12" to "3/11."</p> <p>Issue 7597: Global edit. Changed max length of data sent using LE Set Extended Advertising Data and LE Set Extended Scan Response Data from "252" to "251" in test steps and tables for test case TP/DDI/ADV/BV-24-C, 25-C, 27-C, and 29-C – 32-C.</p> <p>Issue 7601: Added new section and new TCMT entry for test case TP/DDI/ADV/BV-35-C.</p> <p>Issue 7602: Added "Skip to step 11" to step 9 of test case TP/DDI/ADV/BV-24-C.</p> <p>Issue 7615: Changed property value for Round 1 from "0x0040" to "0x0000" in table in test case TP/DDI/ADV/BV-24-C. Changed max length of data for Round 2 from "252" to "31" in table in test case TP/DDI/ADV/BV-24-C and TP/DDI/ADV/BV-25-C.</p> <p>Issue 7618: Changed "AUX_ADV_IND" to "ADV_EXT_IND" in step 7 of test case TP/DDI/ADV/BV-24-C.</p> <p>Issue 7619: Updated test case TP/DDI/ADV/BV-24-C: Changed "IUT stops advertising" to "IUT does not start any additional advertising events" in step 9. Changed "IUT stops advertising within the expected duration" to "IUT does not start any new advertising events after the time specified for Duration has elapsed" in Pass Verdict. Added Notes section.</p> <p>Issue 7622: Updated step 1 in test cases TP/DDI/ADV/BV-19-C – 21-C: Changed "Own_Address_Type[0]" to "Own_Address_Type shall be." Changed "Scanning_Filter_Policy[0]" to "Scanning_Filter_Policy shall be." Updated step 1 in test cases TP/DDI/ADV/BV-23-C – 25-C: Changed "Own_Address_Type[i]" to "Own_Address_Type shall be." Changed "Scanning_Filter_Policy[i]" to "Scanning_Filter_Policy shall be." Updated step 1 in test case TP/CON/INI/BV-13-C: Updated step 1 parameters. Changed "Initiating_Filter_Policy[0]" to "Initiating_Filter_Policy shall be." Changed "Own_Address_Type[0]" to "Own_Address_Type shall be."</p> <p>Issue 7623: Updated step 3 of test case TP/DDI/SCN/BV-23-C - 25-C. Updated table in test case TP/DDI/SCN/BV-23-C and 24-C.</p>
5.0.0r07	2016-09-27	<p>Issue 7572: Updated Pass Verdict for test case TP/DDI/ADV/BV-01-C.</p> <p>Issue 7624: Added 2 steps to the section Common Initial and Final Conditions.</p> <p>Issue 7627: Updated Pass Verdict for test case TP/DDI/ADV/BV-24-C.</p> <p>Issue 7681: Moved "ChSel" bit in sections DDI > Common PDU Contents and CON > Common PDU Contents.</p>



Revision History	Date	Comments
		<p>Issue 7682: Added item HCI 13/13 and feature name "LE Set Minimum Number Of Used Channels Command" to test case TP/FRH/SLA/BV-03-C in TCMT.</p> <p>Issue 7684: Updated test case TP/TIM/SCN/BV-02-C: Added text to introduction, updated MSC, deleted table and references to table in the Test Procedure steps, added Notes. Added new section for test case TP/TIM/SCN/BV-03-C. Added new TCMT entry for test case TP/TIM/SCN/BV-03-C.</p> <p>Issue 7693: Updated table in Test Procedure of test cases TP/DDI/ADV/BV-24-C & 25-C: Added row 7 and column Fragment_Preference. Deleted references to "Fragment_Preference parameter" from test case TP/DDI/ADV/BV-29-C – 32-C.</p> <p>Issue 7697: Updated test case TP/DDI/ADV/BV-25-C: MSC updated. Added coverage of scan response notifications to Test Procedure. Added new step 12. Steps 14–16 updated accordingly. Added new column to table: "Scan_Response_Notification_Enable". New condition added to Pass Verdict.</p> <p>Issue 7705: Updated tests to allow enabling/disabling multiple sets for LE Set Extended Advertising Enable command. Replaced "a specified advertising set and duration" with "Duration[0]" for test case TP/DDI/ADV/BV-22-C. Changed "Duration" to "Duration[0]" for test cases TP/DDI/ADV/BV-24-C – 28-C, TP/DDI/ADV/BV-34-C, TP/CON/ADV/BV-05-C & 06-C, and TP/TIM/ADV/BV-03-C & 04-C. Updated steps 6 and 11 for test cases TP/DDI/ADV/BV-29-C – 32-C. Updated step 4 for test case TP/DDI/ADV/BV-33-C. Updated both MSCs for test case TP/DDI/ADV/BV-31-C.</p> <p>Issue 7718: Updated steps 10 and 12 of test case TP/DDI/ADV/BV-33-C.</p> <p>Issue 7721: Added parameters for Direct Address Type and Direct Address in LE Extended Advertising Report for test cases TP/DDI/SCN/BV-19-C and 20-C.</p> <p>Issue 7722: Updated test case TP/DDI/SCN/BV-19-C: Added scan timeout coverage. Updated step 2. Added new step 5. Updated steps 8 and 9 accordingly. Added new column (LE Set Extended Scan Enable) and new row (16) to table. Added new condition to Pass Verdict.</p> <p>Issue 7728: Deleted test case TP/FRH/SLA/BV-03-C and its corresponding TCMT reference.</p>
5.0.0r08	2016-09-29	<p>Issue 7751: Added repeating step to step 3 for test cases TP/DDI/SCN/BV-19-C, 20-C, 23-C, and 24-C.</p> <p>Issue 7769: Changed "TP/CON/INI/BV-14-C" to "TP/CON/INI/BV-13-C" in TCMT.</p> <p>Issue 7770: "ADV_IND" PDU changed to "ADV_DIRECT_IND" PDU in steps 2 and 3 of test case TP/CON/INI/BV-15-C.</p>



Revision History	Date	Comments
5.0.0r09	2016-10-11	<p>TSE 6899: Added new section "Data Fragmentation Over HCI". Added reference to Section 4.1.7 in Pass verdict for test cases TP/CON/SLA/BV-04-C – 06-C, TP/CON/MAS/BV-03-C – 05-C, and TP/SEC/SLA/BV-05-C. Modified test procedure for test cases TP/CON/MAS/BV-03-C – 05-C and TP/CON/SLA/BV-05-C. Global edits: 1) Changed "1000 data packets" to "100 data packets" (i.e. 100 data packets = 1000 data bytes). 2) Deleted all references to "LL Defragmentation"</p> <p>TSE 6969: Updated MSC and modified Test Procedures and Pass Verdicts for test cases TP/SEC/SLA/BV-06-C, TP/SEC/SLA/BV-09-C, TP/SEC/SLA/BV-10-C, TP/SEC/MAS/BV-06-C, TP/SEC/MAS/BV-09-C, and TP/SEC/MAS/BV-10-C.</p> <p>TSE 7057: Test cases TP/SEC/SLA/BV-06-C, TP/SEC/SLA/BV-08-C – 10-C remapped to LL 1/4 AND LL 9/2 (LE Ping Procedure); test cases TP/SEC/MAS/BV-06-C, and TP/SEC/MAS/BV-08-C – 10-C remapped to LL 1/5 AND LL 9/2 (LE Ping Procedure).</p> <p>TSE 7138: Standby State Preamble Steps updated: five MSCs replaced with one.</p> <p>TSE 7150: Corrected numbering in steps 4 and 9 for test case TP/SEC/SLA/BI-01-C.</p> <p>TSE 7242: Updated MSC and modified Test Procedure and Pass Verdict for test case TP/CON/SLA/BV-26-C.</p> <p>TSE 7291: Added "LL 4/6" to mapping for test case TP/DDI/SCN/BV-14-C.</p> <p>TSE 7438: Updated MSCs and Test Procedures ("connection termination" step) for TP/CON/INI/BV-09-C and TP/SEC/ADV/BV-07-C – 13-C. Updated last step of Test Procedure for TP/CON/INI/BV-10-C – 12-C. Updated MSC for TP/SEC/ADV/BV-06-C. Deleted second to last step of Test Procedure for TP/SEC/ADV/BV-08-C.</p> <p>TSE 7463: Updated Test Procedures steps for TP/DDI/SCN/BV-16-C – 18-C. Added new step 1 to Test Procedure. References to other steps in same Test Procedure updated accordingly. Updated MSC for TP/DDI/SCN/BV-16-C: Modified RPA Timeout position and start/end points. Updated MSC for TP/DDI/SCN/BV-18-C: Changed "(Random_Address)" in first line to "(Random_Static_Address)".</p> <p>TSE 7575: Delete "Backoff Procedure" from Test Suite Structure figure.</p> <p>TSE 7582: Global edit. Changed "advertisement packets" to "advertising packets."</p> <p>TSE 7589: Modified Initial Condition. Changed "common connection interval, up to LL_slave_connSlaveLatency_MAX" to "connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX" for test cases</p>



Revision History	Date	Comments
		<p>TP/CON/SLA/BV-24-C – 29-C, TP/CON/SLA/BV-34-C, TP/CON/MAS/BV-24-C – 26-C, TP/CON/MAS/BV-28-C – 30-C, TP/CON/MAS/BV-34-C, and TP/CON/MAS/BV-35-C.</p> <p>TSE 7596: Updated MSC and Test Procedure for TP/SEC/ADV/BV-04-C and TP/SEC/ADV/BV-06-C.</p> <p>TP/SEC/ADV/BV-06-C MSC updates: Deleted first two commands ("HCI_LE_Add_Device_To_Resolving_List" and "HCI_LE_Set_Address_Resolution_Enable") from Upper Tester to IUT and responses (HCI_Command_Complete_Event) from IUT to Upper Tester. Changed "Executed entire Procedure 4 times" to "...3 times" and deleted "4. A Resolvable Private Address" from configuration options.</p> <p>TSE 6693: Global edit. Modified PDUs in applicable body text and MSCs: Changed "LL_CONNECTION_UPDATE_REQ" to "LL_CONNECTION_UPDATE_IND"; "LL_CHANNEL_MAP_REQ" to "LL_CHANNEL_MAP_IND"; "LL_REJECT_IND_EXT" to "LL_REJECT_EXT_IND"</p>
5.0.0r10	2016-11-02	<p>Issue 7844: Updated Test Procedure for TP/DDI/SCN/BV-21-C and -25-C.</p> <p>Issue 7845: Added reference to Bluetooth Core Specification v5.0. Global edits: For new test cases in Bluetooth Core Specification v5.0, updated cross-references from [3] to [10]; updated TBD cross-references from [x] to [10] (Bluetooth Core Specification v5.0); updated TBD section number in cross-reference for Bluetooth Core Specification v5.0.</p> <p>Issue 7848: Updated Test Procedure for TP/DDI/SCN/BV-21-C. Advertising_SID set to value in event report.</p> <p>Issue 7950: Deleted "Duration" and "all" from Test Procedure in TP/DDI/SCN/BV-19-C.</p>
5.0.0r11	2016-11-10	<p>Issue 8002: In the SCN test procedure, update Step 9 to identify received events as HCI LE Periodic Report events.</p> <p>TSE 7812: Updated Test Procedure step 6 for TP/DDI/SCN/BV-18-C. Added new test case TP/DDI/SCN/BI-03-C. Updated TCMT with test case TP/DDI/SCN/BI-03-C.</p> <p>Issue 8033: Updated TP/DDI/ADV/BV-28-C, Test Procedure, Step 7. In Test Procedure table, changed the Round 2 data length in "HCI_LE_Set_Extended_Advertising_Data (Step 4)" from "0" to "1". Updated figure in TP/DDI/ADV/BV-28-C. Added new Step 2 to the Test Procedure in TP/DDI/ADV/BV-29-C.</p> <p>Issue 8065: Updated tests for periodic advertising scanning - TP/DDI/SCN/BV-21-C [Extended Scanning, Periodic Advertising Reception] and TP/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)] - to set Unused to 0xFF for the LE Periodic Advertising Create Sync command and LE Periodic Advertising Report events.</p>



Revision History	Date	Comments
		<p>Issue 8067: In TP/MAS/BV-41-C, corrected column headers for the RX_PHYS and TX_PHYS parameters (LL_PHY_RSP) for case variations for LE 2M PHY, LE Coded PHY, and LE 2M and LE Coded PHYs.</p> <p>TSE 7833: Modified test case name for TP/DDI/SCN/BV-13-C – 18-C, TP/CON/INI/BV-08-C – 12-C., and TP/SEC/ADV/BV-04-C – 14-C. Added new test cases TP/DDI/SCN/BV-26-C – 29-C, TP/CON/INI/BV-18-C – 21-C, and TP/SEC/ADV/BV-15-C – 20-C. Updated TCMT: Added new test cases TP/SEC/ADV/BV-15-C – 20-C, TP/CON/INI/BV-18-C – 21-C, and TP/DDI/SCN/BV-26-C – 29-C. Deleted test cases TP/DDI/SCN/BV-27-C [Network Privacy – Active Scanning, Local IRK, Peer IRK, Ignore Identity Address] and TP/DDI/SCN/BV-29-C [Device Privacy – Active Scanning, Local IRK, Peer IRK, Accept Identity Address] and their entries in the TCMT.</p> <p>Issue 8066: TP/CON/SLA/BV-40-C: In the test procedure, updated Steps 4a and 4b to identify renamed table values M_TO_S_PHY_LTPREF and S_TO_M_PHY_LTPREF for the M_TO_S_PHY and S_TO_M_PHY variables; under Step 9, renamed parameters to M_TO_S_PHY_LTPREF and S_TO_M_PHY_LTPREF and standardized Case heading to "Lower Tester preferences" in the three tables of test cases (Tables 4.28 – 4.30).</p> <p>TP/CON/SLA/BV-42-C: In the test procedure: In steps 7a and 7b, updated references to renamed M_TO_S_PHY_LTPREF and S_TO_M_PHY_LTPREF parameters and "Lower Tester preference" Case header; under Step 12, renamed parameters to M_TO_S_PHY_LTPREF and S_TO_M_PHY_LTPREF and standardized Case header to "Lower Tester preferences" in the three tables (Tables 4.31 – 4.33).</p>
5.0.0	2016-12-13	Approved by BTI. Prepared for TCRL 2016-2 publication.
5.0.1r00	2017-03-01	<p>TSE 8656: Changed "HCI_LE_Encryption_Change" to "HCI_Encryption_Change" in test cases TP/SEC/MAS/BV-01 and TP/SEC/MAS/BV-04.</p> <p>TSE 7854: Changed TP/SEC/ADV/BV-03-C Initial condition from "Non-connectable Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map)" to "Device Address Set (supported type of address, any address)". Added "according to IUT address set in initial condition" to the end of step 3 test procedure for figure 4.284 in TP/SEC/ADV/BV-03-C.</p> <p>Not part of TSE but renumbered test procedure steps for figure 4.284 in TP/SEC/ADV/BV-03-C.</p> <p>TSE 7843: added an HCI_LE_Set_Resolvable_Private_Address_Timeout to the MSC for TP/SEC/ADV/BV-13-C.</p>



Revision History	Date	Comments
		<p>TSE 7852: Added new steps 9-11 to TP/SEC/ADV/BV-12-C and updated MSC accordingly. Added HCI Disconnect in final sequence of MSC for TP/SEC/ADV/BV-13-C.</p> <p>TSE 7856: In the TP/CON/MAS/BV-07-C MSC, moved HCI_LE_Connection_Update_Complete_Event to the end of the sequence and fixed the typo in "_Complete_".</p> <p>TSE 7860: clarified Own_Address_Type requirements in TP/SEC/ADV/BV-13-C and added RPA requirement in initA address to pass verdict.</p> <p>TSE 8158: expanded step 1 of TP/CON/MAS/BV-43-C to account for the full scope of required and allowed behavior. Updated MSC accordingly.</p> <p>TSE 8304: added "or until IUT sends a SCN_RSP" to step 6 in parts a and b of test procedure of TP/DDI/ADV/BV-05-C.</p>
5.0.1r01	2017-03-08	<p>TSE 8293: Updates to test procedures and MSCs to clarify timing and deviation in TP/CON/INI/BV-01-C, TP/CON/INI/BV-13-C, TP/CON/INI/BV-14-C, TP/CON/INI/BV-15-C, TP/CON/INI/BV-16-C, TP/CON/INI/BV-17-C</p> <p>TSE 8327: TP/CON/INI/BV-20-C - corrected step reference to step 5</p> <p>TSE 8345: In MSC for TP/CON/SLA/BV-57-C, changed LL_PHY_REQ Both Directions = 1Ms/s to LL_PHY_REQ Both Directions = LE Coded PHY</p> <p>TSE 8348: Fixed difference between MSC and Test Procedure text in TP/CON/SLA/BV-57-C and TP/CON/MAS/BV-55-C</p> <p>TSE 8369: added MSC figure to TP/CON/SLA/BV-42-C</p> <p>TSE 8374: corrected "LL_PHY_UPDATE_PEQ" to "LL_PHY_UPDATE_IND" in TP/CON/MAS/BV-49-C, TP/CON/MAS/BV-50-C, and TP/CON/MAS/BV-52-C.</p> <p>TSE 8376: Corrected "expects with" to "expects" in TP/CON/MAS/BV-50-C, TP/CON/MAS/BV-54-C, and TP/CON/MAS/BV-55-C</p> <p>TSE 8377: Corrected typo in Step 8 of TP/CON/SLA/BV-58-C.</p> <p>TSE 8378: In TP/CON/SLA/BV-54-C MSC, updated the first "2Ms/s to "LE Coded PHY". In TP/CON/SLA/BV-57-C MSC part B, "1Ms/s" updated to "LE Coded PHY".</p> <p>TSE 8379: Deleted HCI_LE_Set_PHY and HCI_Command_Status_Event at start of Part B of MSC in TP/CON/MAS/BV-54-C and TP/CON/MAS/BV-55-C.</p> <p>TSE 8380: Removed unnecessary HCI_LE_Set_PHY from TP/CON/SLA/BV-51-C.</p> <p>TSE 8383: Corrected formatting error in "MAS" heading for Section 4.3.5.</p> <p>TSE 8388: Corrected typo in TP/DDI/ADV/BV-27-C.</p> <p>TSE 8390: Deleted "On Round 1 only..." in step 1 of TP/DDI/ADV/BV-27-C.</p>



Revision History	Date	Comments
		<p>TSE 8413: moved "...that was used in the SCAN_REQ packet" from step 6 to step 7 in TP/DDI/SCN/BV-18-C</p> <p>TSE 8416: Replaced 0xFF with 0x00 for Unused value in TP/DDI/SCN/BV-21-C, TP/DDI/SCN/BV-25-C.</p> <p>TSE 8468: In Section 4.1.6, added the recommendation that the control of the IUT is arranged for testing purposes so that Inconclusive verdicts are the exceptions rather than the rule. TSE 8587: In TP/DDI/ADV/BV-25-C, corrected scan data length in rounds 1 and 9 from 0 to 1.</p>
5.0.1r02	2017-03-15	<p>TSE 8384: In TP/CON/MAS/BV-51-C: Deleted "LE Coded" from the title. In the MSC, changed "0x02" to "0x02 or 0x04" in two places (first step). In step 1, changed: 1) "zero and" to "zero," and 2) "prefer a PHY" to "prefer a single supported PHY". In steps 3a/b, changed 1) "the opposite bit in" to "a different bit in the"; 2) change "both bits" to "more than one bit". Appended to step 3a: "The bit set must correspond to a PHY that the IUT supports." In TCMT, remapped TP/CON/MAS/BV-51-C to feature "PHY Update Procedure (Master)."</p> <p>TSE 8389: In TP/DDI/ADV/BV-27-C: Modified step 7: Deleted "On Round 1, or if the Advertising Data ID has changed since the last round (N-1)," and the final paragraph, "If the Advertising Data ID is the same as the last round, skip to Step 10." Added new step to Test Procedure. In step 10, changed "6–8" to "6–9". In the Pass Verdict: 1) Deleted from the third bullet: "If advertising data was specified for the round"; 2) added fourth bullet: "The Advertising Data ID changes whenever the data changes."</p>
5.0.1r03	2017-03-16	<p>TSE 8222: Various fixes in TP/CON/ADV/BV-05-C and 06-C to correct transition from Initiating State to Connection State as well as other procedural corrections</p> <p>TSE 7859: TP/SEC/ADV/BV-05-C: In the MSC, for HCI_LE_Set_Advertising_Parameters, add Adv_filter_policy =0x01; TP/SEC/ADV/BV-07-C: In the MSC, for HCI_LE_Set_Advertising_Parameters, add Adv_filter_policy =0x02</p> <p>TSE 7879: Removed step 7, added text to step 3 and updated numbering in step 15 of TP/CON/INI/BV-12-C to clarify IUT address type.</p> <p>TSE 8290: TP/CON/ADV/BV-07-C – 10-C, TP/DDI/ADV/BV-06-C-07-C, TP/DDI/ADV/BV-09-C: Clarified advertising interval time wording in test procedures and in MSCs where needed.</p> <p>TSE 8305: Updates to interval length in test procedure and MSC for TP/DDI/ADV/BV-11-C.</p> <p>TSE 8320: Corrected the Advertising_SID value in step 1 of TP/DDI/ADV/BV-35-C.</p> <p>TSE 8335: LeSetPhy Param ALLPHYS corrected to 0x03 instead of 0x05 in TP/CON/SLA/BV-54-C.</p>



Revision History	Date	Comments
		<p>TSE 8341: Completed PDU name changes from TSE 6693: "LL_CONNECTION_UPDATE_REQ" to "LL_CONNECTION_UPDATE_IND"; "LL_CHANNEL_MAP_REQ" to "LL_CHANNEL_MAP_IND"; "LL_REJECT_IND_EXT" to "LL_REJECT_EXT_IND". Also corrected PDU names without "LL_" in front (e.g. "REJECT_EXT_IND")</p> <p>TSE 8401: Editorial and clarifying changes to TP/CON/SLA/BV-42-C: In step 2, change "will all fields" to "with all fields". In step 3, change "Lower" to "Upper". After step 8 add "If both the M_TO_S_PHY and S_TO_M_PHY fields of the LL_PHY_UPDATE_IND are zero, skip to step 11." In step 10, delete "(or no change ... PDU)". Change the Inconclusive Verdict to: "The PHY does not change (equivalently, steps 9 and 10 are not carried out) at least once during this test case because of the rules in step 7."</p> <p>TSE 8418: Clarification on skip parameters in test procedure and pass verdict of TP/DDI/SCN/BV-21-C</p> <p>TSE 8453: Replaced part B of MSCs in TP/CON/SLA/BV-53-C and TP/CON/SLA/BV-59-C, removing all messages following the LL_PHY_UPDATE_IND (No Change).</p> <p>TSE 8538: Advertising interval clarification in test procedure of TP/DDI/SCN/BV-16-C</p> <p>TSE 8549: Corrected scan interval settings for multiple PHY in TP/DDI/SCN/BV-23-C – 25-C.</p> <p>TSE 8566: Various fixes to TP/CON/ADV/BV-05-C and 06-C test procedure and MSCs related to the advertising disable step.</p>
5.0.1r04	2017-04-18	Integrated review comments on TSEs 8566, 8222, 8335, 8293, 8260, 7860, 8320, 8348, 8305, 8369, 7852, 8327, 8290
5.0.1r05	2017-05-09	<p>TSE 8264: LE Coded already removed from the title of TP/CON/MAS/BV-51-C from TSE 8384. In the MSC Figure 4.237, changed "0x02 and 0x04" to "Other than 1Mbit" for RX_PHYS and TX_PHYS.</p> <p>TSE 8277: Various fixes for TP/DDI/SCN/BV-20-C: Added a new step 1 and renumbered the test procedures according to the new numbering. Removed "Repeat steps 1-7 for each Round shown in Table 4.16." from the whole test procedure. Updated test procedure step 9 and 10 to include "only rounds 5-8 in Table 4.16 are executed, in Step 2" and removed "and in Step 8 only rounds 5-10 in Table 4.16 are executed". Changed the scan data for Table 4.16 from None to "1" for Round 5, 6 and 7. Changed from Step 3 to "Step 4" in pass verdict.</p> <p>TSE 8289: Removed 2 text boxes "It is checked that no packets....". All instances of LL_CONNECTION_UPDATE_REQ are replaced by LL_CONNECTION_UPDATE_IND. Fixed spelling of "transmission" from MSC for TP/TIM/SLA/BV-01-C.</p>



Revision History	Date	Comments
		<p>Removed text box "It is checked that no packets...." from MSC for TP/TIM/SLA/BV-02-C and TP/TIM/SLA/BV-03-C.</p> <p>TSE 8323: Revised the first part of step 11 test procedure and replaced "PHY bit in the LL_PHY_RSP" with "bit in the TX_PHYS field" for the last inconclusive verdict for TP/CON/SLA/BV-53-C.</p> <p>Revised the first part of step 8 test procedure and replaced "allow the Lower Tester to select the LE Coded PHY" with "set the LE Coded PHY bit in the TX_PHYS field" for the last inconclusive verdict for TP/CON/SLA/BV-59-C.</p> <p>TSE 8362: Added dotted arrow for AUX-ADV_IND to MSC Figure 4.63 and updated step 4 of test procedure for TP/DDI/ADV/BV-34-C.</p> <p>TSE 8579: Various fixes to TP/CON/MAS/BV-50-C: Removed steps 1 and 2 from the test procedure of TP/CON/MAS/BV-50-C and renumbered steps in test procedure, pass verdict, and inconclusive verdict to match new numbering. Updated MSC Figure 4.235 to reflect removal of step 1 and 2 from test procedure.</p> <p>TSE 8584: Corrected "andif" to "and if" in Pass/Inconclusive/Fail Verdict Conventions section.</p> <p>TSE 8586: Replaced line 3 of pass verdict with "The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms," for TP/DDI/ADV/BV-15-C.</p> <p>TSE 8860: Removed (Identity Address) from SCAN_RSP in MSC Figure 4.299 and changed part of pass verdict from "Lower Tester's device identity address" to "the advertiser RPA" for TP/SEC/ADV/BV-18-C.</p> <p>TSE 8895: Changed title of TP/SEC/MAS/BV-13-C from "FEATURES" to "FEATURE".</p> <p>TSE 8844: Added clarifying text about own address type "(0x02 or 0x03) to initial condition of TP/DDI/ADV/BV-09-C, TP/DDI/SCN/BV-13-C, TP/DDI/SCN/BV-14-C, TP/DDI/SCN/BV-26-C.</p> <p>TSE 8681: Renumbered test procedure steps to start at 1 for TP/SEC/ADV/BV-03-C.</p> <p>TSE 8548: Updated MSC Figure 4.296 with "Advertising_Filter_Policy" and added text "Advertising Filter Policy shall be set to 0x01" to step 3 of the test procedure for TP/SEC/ADV/BV-15-C.</p>
5.0.1r06	2017-05-15	<p>TSE 8772: Updated Peer_Addr_Type from 2 to 0x0 in MSC for TP/CON/INI/BV-09-C, TP/CON/INI/BV-10-C, TP/CON/INI/BV-11-C, TP/CON/INI/BV-12-C, TP/CON/INI/BV-18-C, TP/CON/INI/BV-19-C, TP/CON/INI/BV-20-C, TP/CON/INI/BV-21-C.</p> <p>TSE 8693: Changed Maximum_Advertising_Data_Length value from 0xFFFF to 0x0672 in the test procedure and pass verdict for TP/DDI/ADV/BV-24-C, TP/DDI/ADV/BV-25-C, TP/DDI/ADV/BV-</p>



Revision History	Date	Comments
		<p>26-C, TP/DDI/ADV/BV-27-C, TP/DDI/ADV/BV-29-C, TP/DDI/ADV/BV-30-C, TP/DDI/ADV/BV-31-C, TP/DDI/ADV/BV-32-C.</p> <p>TSE 9040: Replaced 2µs with 1.5µs in MSC and in step 2 of the test procedure, and added notes to the pass verdict for TP/TIM/MAS/BV-01-C, TP/TIM/MAS/BV-02-C.</p> <p>TSE 8523: Various fixes to TP/DDI/ADV/BV-33-C: Updated MSC to show the enable command being executed only once, added text to allow for an inconclusive verdict when a set cannot be created to step 1 and 2 of the test procedure, Updated step 5 to specify that the Lower Tester scans for each adv. PHY supported by the IUT, Clarified need to be repeated for each “first set” to steps 9-11 in the test procedure, corrected the pass verdict to specify conditions for each adv. set, rather than for each round (since the test does not have rounds), added inconclusive verdict to the expected outcome, and added TP/DDI/ADV/BV-33-C to table 4.2.</p>
5.0.1r07	2017-05-18	<p>TSE 7843: Added a new paragraph (“The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.”) under the Initial Condition section for the following test cases: TP/SEC/ADV/BV-03-C, TP/SEC/ADV/BV-05-C, TP/SEC/ADV/BV-07-C, TP/SEC/ADV/BV-08-C, TP/SEC/ADV/BV-10-C, TP/SEC/ADV/BV-11-C, TP/SEC/ADV/BV-12-C, TP/SEC/ADV/BV-13-C, TP/SEC/ADV/BV-14-C, TP/SEC/ADV/BV-15-C, TP/SEC/ADV/BV-16-C, TP/SEC/ADV/BV-17-C, TP/SEC/ADV/BV-18-C, TP/SEC/ADV/BV-19-C, TP/SEC/ADV/BV-20-C</p> <p>Changed "HCl_LE_Set_Random_Private_Address_Timeout" to "HCl_LE_Set_Resolvable_Private_Address_Timeout" in the MSCs for the following test cases:</p> <p>TP/DDI/SCN/BV-13-C, TP/DDI/SCN/BV-16-C, TP/DDI/SCN/BV-18-C, TP/SEC/ADV/BV-07-C, TP/SEC/ADV/BV-08-C, TP/SEC/ADV/BV-10-C</p> <p>*Related TSEs: 6491 and 6655</p> <p>TSE 8304: For TP/DDI/ADV/BV-05-C, corrected step references in steps 20, 26, and 28.</p>
5.0.1r08	2017-05-21	<p>TSE 8381: added new paragraph for LE PHY Update Procedure in Section 4.1.5.7.3, Optional Test Steps; corrected handling of PHY update procedures in TP/CON/MAS/BV-43-C, TP/CON/MAS/BV-49-C, TP/CON/MAS/BV-50-C, TP/CON/MAS/BV-52-C, TP/CON/MAS/BV-53-C, TP/CON/MAS/BV-54-C, TP/CON/MAS/BV-55-C, TP/CON/SLA/BV-42-C, TP/CON/SLA/BV-47-C, TP/CON/SLA/BV-49-C, TP/CON/SLA/BV-50-C, TP/CON/SLA/BV-52-C, TP/CON/SLA/BV-53-C, TP/CON/SLA/BV-54-C, TP/CON/SLA/BV-55-C,</p>



Revision History	Date	Comments
		TP/CON/SLA/BV-56-C, TP/CON/SLA/BV-57-C, TP/CON/SLA/BV-58-C, TP/CON/SLA/BV-59-C
5.0.1r09	2017-05-23	<p>TSE 8386: Clarified checking of AdvA / TargetA fields in TP/DDI/ADV/BV-24-C – 34-C and added section 4.2.1.2 on Extended Advertising.</p> <p>TSE 8530: Clarified order of procedures with changes to steps 7-8 of TP/CON/MAS/BV-28-C.</p> <p>TSE 8532: added clarification of behavior based on Max_Extended_Advertising_Events parameter value via changes to test procedure steps, reference tables, and Pass verdict in TP/DDI/ADV/BV-24-C.</p> <p>TSE 8542: Replaced “Undirected Advertising (selected Adv_Interval, selected Adv_Interval...)” with “Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max...)” (67 instances); affected test cases: TP/DDI/ADV/BV-01-C – 09-C, TP/DDI/ADV/BV-15-C – 35-C, TP/DDI/ADV/BI-01-C – 02-C, TP/CON/ADV/BV-01-C – 03-C, TP/CON/ADV/BV-05-C – 10-C, TP/CON/ADV/BI-01-C, TP/TIM/ADV/BV-01-C – 02-C, TP/TIM/SLA/BV-02-C – 03-C, TP/ENC/ADV/BI-01-C – 02-C, TP/FRH/ADV/BV-01-C, TP/SEC/ADV/BV-01-C – 19-C. Additional editorial changes.</p> <p>TSE 8666: Added missing step 2 to test cases TP/DDI/ADV/BV-07-C – 09-C and updated subsequent step references.</p> <p>TSE 8568: Clarified advertising interval length in test procedure steps of TP/DDI/ADV/BV-24-C – 35-C.</p> <p>TSE 8665: Added steps to TP/DDI/ADV/BV-29-C and 32-C to clarify disabling and clearing of advertising sets.</p> <p>TSE 8706: Added clause regarding advertising channel use for the various PHY to step 1 of TP/ADV/DDI/BV-33-C.</p> <p>TSE 8822: Extensive changes to test procedures in TP/FRH/MAS/BV-01-C and 02-C. Replaced MSCs. Added Inconclusive verdicts and updated Table 4.2 accordingly.</p> <p>TSE 8662: Extensive changes through steps 5-15 to clarify TP/DDI/SCN/BV-25-C. Corresponding updates to MSC.</p> <p>TSE 8866: Replaced Part A of MSC and added clause in test procedure to correct IUT command complete response to set data length command. Updated in: TP/CON/MAS/BV-49-C, TP/CON/MAS/BV-50-C, TP/CON/MAS/BV-52-C, TP/CON/MAS/BV-53-C, TP/CON/MAS/BV-54-C, TP/CON/SLA/BV-49-C, TP/CON/SLA/BV-50-C, TP/CON/SLA/BV-52-C , TP/CON/SLA/BV-53-C, TP/CON/SLA/BV-54-C, TP/CON/SLA/BV-55-C, TP/CON/SLA/BV-56-C, TP/CON/SLA/BV-58-C, TP/CON/SLA/BV-59-C.</p>
5.0.1r10	2017-06-02	TSE 8375: Global edit to the Common Initial and Final Conditions section and test cases LL/CON/SLA/BV-49-C, LL/CON/SLA/BV-50-C, LL/CON/SLA/BV-52-C, LL/CON/SLA/BV-53-C,



Revision History	Date	Comments
		LL/CON/MAS/BV-49-C, and LL/CON/MAS/BV-50-C: Changed all references to "1Ms/s" to "LE 1M PHY"; changed all references to "2Ms/s" to "LE 2M PHY". Global edit. Converted legacy TCIDs to new TCID convention: Changed "TP/" to "LL".
5.0.1r11	2017-06-08	Editorial changes per review comments received against several TSEs (added in real time on BTI call) REMOVE THIS REV HISTORY LINE BEFORE PUBLICATION
5.0.1	2017-07-05	Approved by BTI. Prepared for TCRL 2017-1 publication.
5.0.2r00	2017-07-20	TSE 8580: Replace "Repeat for up to 20 advertising intervals[...]" by "Repeat for at least 20 advertising intervals[...]" in the following test cases. LL/DDI/SCN/BV-15-C, 16-C, 17-C, 18-C, 19-C, 20-C, 23-C, 24-C and LL/DDI/SCN/BI-03-C TSE 8865: For test case LL/CON/MAS/BV-13-C: Change "The tester acts in the slave role [...]" to "The Lower Tester acts in the slave role [...]." In Test Procedure step 2, changed name to match 5.0 spec. In step 3, revised for clarity and revised incorrect reference of REQ_PDU. In step 4, deleted extra use of "event" and changed name to match 5.0 spec. Deleted step 5. Updated the Pass Verdict. Added Note to cover the name update. Revised MSC. Updated to HCI_LE_Read_Remote_Features to match BT 5.0 Core Spec. Changed "LL_SLAVE_FEATURE_REQ" to "LL_FEATURE_REQ" in MSC note. TSE 8865 continued: For test case LL/SEC/MAS/BV-11-C: Added LL_ prefix to REJECT_EXT_IND. Updated MSC to remove the incorrect use of "LL_SLAVE_FEATURE_REQ" by master. In Test Procedure, deleted step 1. In step 2, added LL_ prefix. In step 3, changed formatting to match PDU variable name. In step 4, changed to match the pass verdict (i.e. error code becomes 'status parameter' since the HCI event doesn't have a variable named error code.) Updated text in Pass Verdict.
5.0.2r01	2017-08-16	TSE 9600: Updated step number in test procedure for LL/CON/ADV/BV-06-C. TSE 9401: For LL/CON/MAS/BV-55-C, removed optional PHY update procedure from MSC. TSE 9403: For LL/CON/ADV/BV-06-C, deleted step 14; reordered bullets of the Pass Verdict by step number; and added Fail Verdict.



Revision History	Date	Comments
		TSE 9431: For LL/CON/SLA/BV-14-C, changed all instances of "Features_Req packet" to "LL_FEATURE_REQ PDU" and "Features_Rsp" to "LL_FEATURE_RSP". For LL/CON/MAS/BV-19-C, changed all instances of "Features_Req packet" to "LL_FEATURE_REQ PDU" and "Features_Rsp packet" to "LL_FEATURE_RSP PDU". Revised initial condition for LL/CON/SLA/BV-14-C and revised test procedure and Figure 4.206 for LL/CON/MAS/BV-19-C.
5.0.2r02	2017-08-18	TSE 8723: For LL/DDI/SCN/BV-11-C, updated introduction, initial condition, prior steps, and pass verdict text; replaced figure. TSE 9394: For LL/SEC/MAS/BV-06-C, LL/SEC/MAS/BV-07-C, LL/SEC/MAS/BV-09-C, LL/SEC/SLA/BV-06-C, LL/SEC/SLA/BV-07-C, and LL/SEC/SLA/BV-09-C, removed Lower Tester support statements from initial conditions.
5.0.2r03	2017-08-22	TSE 9402: For LL/CON/ADV/BV-05-C, revised MSC boxes. TSE 9602: Revised test procedure values in LL/DDI/ADV/BV-26-C. TSE 9613: For LL/DDI/SCN/BV-25-C, revised Test Procedure step 6.d. TSE 9707: Deleted LL/CON/SLA/BV-41-C from the table in Pass/Inconclusive/Fail Verdict Conventions. TSE 9723: Fixed typo in LL/SEC/MAS/BV-06-C Test Procedure text.
5.0.2r04	2017-08-23	TSE 9666: For LL/DDI/ADV/BV-24-C - ...27-C, and 29-C - ...32-C, changed the minimum value of Maximum_Advertising_Data_Length from 0x00BF to 0x001F.
5.0.2r05	2017-08-28	TSE 9379: Updated step 1 and Figure 4.191 for LL/CON/SLA/BI-09-C. TSE 9404: For LL/DDI/SCN/BV-11-C and LL/CON/INI/BV-07-C, revised text and removed references to advertising data ADV_DIRECT_IND packets. TSE 9573: Revised pass verdict text for LL/CON/MAS/BV-41-C. TSE 9476: Corrected mapping for LL/CON/SLA/BV-04-C - ...06-C and LL/CON/MAS/BV-03-C - ...05-C.
5.0.2r06	2017-09-01	TSE 8746: Added new test cases LL/PAC/SLA/BI-01-C and LL/PAC/MAS/BI-01-C to the PAC section and the TCMT.
5.0.2r07	2017-09-15	TSE 8797: Revised text and figures for test case LL/DDI/ADV/BV-33-C. Updated Test Procedure step 1 in LL/DDI/ADV/BV-33-C. TSE 8536: Corrected Duration in LL/DDI/SCN/BV-19-C test procedure step 3.
5.0.2r08	2017-09-19	AoA/AoD CR incorporation: Updated sections: Test Strategy; Pass/Inconclusive/Fail Verdict Conventions; TCMT; LL/DDI/SCN/BV-21-C, LL/DDI/SCN/BV-25-C; New sections: LL/DDI/ADV/BV-36-C -37-C, LL/DDI/SCN/BV-29-



Revision History	Date	Comments
		C – 32-C, LL/CON/SLA/BV-60-C – 76-C, LL/CON/MAS/BV-56-C – 72-C.
5.0.2r09	2017-09-26	TSE 9217: Revised privacy requirements in initial conditions and figures in test cases LL/DDI/SCN/BV-26-C, LL/CON/INI/BV-18-C – 19-C, and LL/SEC/ADV/BV-15-C – 17-C. Revised TCMT for LL/DDI/SCN/BV-28-C. TSE 9804: For LL/CON/ADV/BV-05-C MSC, deleted last two arrows at the end of Part A.
5.0.2r11	2017-10-02	TSE 8671: Revised LL/DDI/SCN/BV-19-C test procedure and pass verdict text.
5.0.2r12	2017-10-09	TSE 9885: Corrected style of PDU error code field to “ErrorCode” (global change).
5.0.2r13	2017-10-10	TSE 9831: For LL/CON/SLA/BV-22-C – 23-C, updated MSC, test procedure, and pass verdict; for LL/CON/MAS/BV-13-C, updated MSC; for LL/CON/MAS/BV-19-C, updated MSC and test procedure. TSE 9708: Revised LL/CON/MAS/BV-43-C test procedure and pass verdict. TSE 9605: Added new subsection, “Data PDUs and Empty PDUs” to the “Introduction” section. TSE 9527: Revised test cases LL/DDI/SCN/BV-23-C – 24-C text, MSC, test procedure, and pass verdict. TSE 9863: Corrected errors introduced by TSE 8416 in LL/DDI/SCN/BV-21-C and LL/DDI/SCN/BV-25-C test procedures. TSE 8320: Revised LL/DDI/ADV/BV-35-C test procedure. TSE 9913: Corrected errors introduced by TSE 8844 in LL/DDI/ADV/BV-09-C and LL/SEC/ADV/BV-09-C initial conditions. TSE 9879: Revised LL/SEC/SCN/BV-01-C MSC.
5.0.2r14	2017-10-11	TSE 9662: Revised inconclusive verdicts table and LL/DDI/ADV/BV-33-C text, initial condition, MSC, and test procedure. TSE 9393: Revised test cases: LL/DDI/ADV/BV-05-C and 17-C text, initial condition, test procedure, MSC, and expected outcome; LL/DDI/ADV/BV-06-C text, initial condition, and test procedure; LL/DDI/ADV/BV-09-C test procedure and expected outcome. TSE 8686: Revised LL/DDI/SCN/BI-03-C initial condition, MSC, test procedure, and expected outcome. TSE 8670: Revised LL/CON/SLA/BI-02-C and LL/CON/MAS/BI-02-C test procedure and expected outcome.
5.0.2r15	2017-10-12	TSE 8100: Revised the “INI” subsection in the “CON” section. Revised MSCs: LL/CON/INI/BV-01-C, 04-C, 08-C – 17-C, 20-C, and 21-C; and LL/CON/INI/BI-02-C.



Revision History	Date	Comments
		TSE 9749: Revised LL/DDI/ADV/BV-06-C, 07-C, and 09-C test procedures and MSCs.
5.0.2r17	2017-10-18	<p>TSE 9930: Revised LL/CON/SLA/BI-04-C text, initial condition, MSC, test procedure, and expected outcome.</p> <p>TSE 9927: Added new subsection “Outstanding Commands Prior to Disconnection” to the “Test Cases (TC),” “Introduction,” section.</p> <p>TSE 9871: Revised LL/SEC/ADV/BV-01-C, LL/SEC/SCN/BV-01-C, and LL/CON/INI/BV-01-C text, test procedure, and expected outcome.</p> <p>TSE 9897: For LL/CON/SLA/BV-40-C: revised test procedure, deleted the “PDU payload contents for each case variation for LE 2M PHY supported and LE Coded PHY not supported” and “PDU payload contents for each case variation for LE Coded PHY supported and LE 2M PHY not supported” tables, and revised the “PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported” table. Updated table numbers.</p> <p>TSE 9729: Deleted LL/CON/MAS/BV-36-C – 40-C and LL/CON/SLA/BV-35-C – 39-C, added new test cases LL/CON/MAS/BV-73-C – 80-C and LL/CON/SLA/BV-77-C – 84-C, updated table of inconclusive verdicts, and updated the TCMT.</p>
5.0.2r18	2017-10-19	<p>TSE 9898: Added new test case LL/DDI/ADV/BV-38-C and added to the TCMT.</p> <p>TSE 9874: Added new test cases LL/CON/MAS/BV-81-C – 82-C and LL/CON/SLA/BV-85-C – 86-C and updated the TCMT.</p> <p>TSE 9901: Added new test case LL/DDI/SCN/BV-33-C and updated the TCMT.</p> <p>TSE 9899: Added new test case LL/DDI/SCN/BV-34-C and updated the TCMT. Revised all TCMT item numbers changed by revisions to the Protocol Scanning Features table in the LL.ICS.</p>
5.0.2	2017-12-07	Approved by BTI. Prepared for TCRL 2017-2 publication.
5.0.3r00-17	2018-01-05 – 2018-06-20	<p>Template update.</p> <p>Issue 10258: Updated Reference and Test Procedure for LL/CON/SLA/BV-65-C, 68-C; LL/CON/MAS/BV-61-C, 64-C; added new section “Both Connected Roles”; added items to TCMT: LL 6/25 to LL/CON/SLA/BV-87-C and LL 7/25 to LL/CON/MAS/BV-83-C.</p> <p>Issue 10212: For LL/DDI/ADV/BV-36-C, 37-C: Added new sentence “Advertisements without data, along with chaining, are tested”; replaced MSC; revised Test Procedure; updated Pass Verdict. For LL/DDI/SCN/BV-29-C, 30-C: Replaced MSC; revised Test Procedure; updated Pass Verdict. For LL/DDI/SCN/BV-31-C, 32-C: updated Test Procedure.</p>



Revision History	Date	Comments
		<p>Issue 10203: Updated Test Procedure for LL/DDI/SCN/BV-29-C, 30-C; LL/CON/SLA/BV-60-C, 67-C, 69-C, 70-C; LL/CON/MAS/BV-56-C, 63-C, 65-C, 66-C.</p> <p>Issue 10180: Added new item LL 11/TBD1 to LL/DDI/SCN/BV-35-C, 36-C to TCMT. Added new test cases LL/DDI/SCN/BV-35-C, 36-C.</p> <p>Issue 10133: Added new item "LL 2/1 AND LL 3/13" to LL/DDI/ADV/BV-39-C to TCMT. Added new test case LL/DDI/ADV/BV-39-C.</p> <p>Issue 10008: Updated Test Procedure for LL/CON/SLA/BV-60-C – 64-C, 67-C; LL/CON/MAS/BV-56-C – 60-C, 63-C. For LL/CON/SLA/BV-69-C, 70-C: Replaced MSC; updated Test Procedure; updated Pass Verdict. For LL/CON/MAS/BV-65-C, 66-C: Replaced MSC; updated Test Procedure; updated Pass Verdict.</p> <p>TSE 10232 (rating 1): Moved Section 4.3.2 (Common PDU Contents) before Section 4.3.1 (ADV).</p> <p>TSE 10204 (rating 1): Corrected Test Procedure for LL/DDI/SCN/BV-34-C.</p> <p>TSE 10355 (rating 1): Replaced MSC and revised Test Procedure for LL/DDI/SCN/BV-19-C.</p> <p>TSE 10186 (rating 1): Added Test Purpose heading and replaced MSC for LL/SEC/ADV/BV-02-C - 03-C.</p> <p>TSE 10167 (rating 3): Revised test procedure step 5 in LL/DDI/ADV/BV-32-C to reject HCI_LE_Set_Extended_Advertising_Data commands only if the Data Length for a given set is greater than zero.</p> <p>TSE 10328 (rating 3): Replaced MSC and revised test procedure step 3 for LL/TIM/SCN/BV-01-C and 02-C to clarify use of T_MAFS.</p> <p>TSE 9975 (rating 3): Added an inconclusive verdict as a possible outcome to the expected outcome for LL/CON/SLA/BV-20-C.</p> <p>TSE 9896 (rating 3): Replaced test procedure steps 7 and 8 for LL/CON/MAS/BV-28-C to make message order more flexible.</p> <p>TSE 9907 (rating 3): Removed mandatory condition from test procedure step 3 in LL/CON/MAS/BV-33-C.</p> <p>TSE 10076 (rating 3): In test purpose, changed "REJECT_EXT_IND" to "LL_REJECT_IND or LL_REJECT_EXT_IND"; in MSC, made LL_REJECT_IND (ErrorCode: 0x06) be an alternative to the LL_REJECT_EXT_IND; in step 5, allowed either PDU to be sent; in expected outcome pass verdict, allowed either PDU to be sent;</p>



Revision History	Date	Comments
		<p>in TCMT, removed the requirement for LL 9/4 for LL/SEC/SLA/BV-11-C.</p> <p>TSE 10151 (rating 3): Revised repetitions from 100 to 10 in MSC; revised test procedure step 9; and added a column for repeat count (step 9) for LL/DDI/ADV/BV-28-C.</p> <p>TSE 10160 (rating 3): Revised test procedure steps 1 and 2 for LL/DDI/ADV/BV-33-C.</p> <p>TSE 10234 (rating 3): Replaced MSC, revised test procedure step 3, and deleted notes to unconditionally allow both 0x16 and 0x22 as valid error codes for HCI_Disconnect_Complete_Event in LL/CON/MAS/BI-02-C and LL/CON/SLA/BI-02-C.</p> <p>TSE 10209 (rating 2): Moved and renumbered LL/DDI/ADV/BV-38-C to LL/DDI/SCN/BV-37-C and revised its corresponding TCMT entry.</p> <p>TSE 9991 and 10285 (rating 2): In TCMT: added LL 9/6 AND to mapping for LL/CON/SLA/BV-49-C, 50-C, 52-C, LL/CON/MAS/BV-49-C, 50-C; changed (LL 9/7 OR 9/9) to LL 9/6a in mapping and added “multiple PHYs” to feature for LL/CON/SLA/BV-44-C, 45-C, 48-C; moved LL/CON/SLA/BV-51-C and LL/CON/SLA/BI-09-C to new mapping LL/1/4 AND LL 6/23a and feature “PHY Update Procedure (Slave)”; changed (LL 9/7 OR 9/9) to LL/9/6a in mapping for LL/CON/SLA/BV-43-C, 46-C, 47-C, LL/CON/MAS/BV-41-C to 46-C, 51-C; added LL 9/6 AND to mapping for LL/CON/MAS/BV-49-C, 50-C; changed LL/6/11 AND (LL 9/7 OR 9/9) to LL 7/11 AND LL 7/23a in mapping for LL/CON/MAS/BV-47-C; changed LL/6/10b AND (LL 9/7 OR LL 9/9) to LL 7/10b AND LL 7/23a in mapping for LL/CON/MAS/BV-48-C; added “Responding to” to feature for LL/CON/SLA/BV-43-C; changed (LL 9/7 OR LL 9/9) AND LL 9/8 to LL 6/23a in mapping and “PHY Update Procedure (Slave)” to “Responding to PHY Update Procedure (Slave)” in feature for LL/CON/SLA/BV-42-C; moved LL/CON/SLA/BV-55-C, 56-C, 58-C, 59-C to new mapping LL 1/4 AND LL 9/6 AND LL 9/9 and feature “PHY Update Procedure (LE Coded PHY, Data Length Extension, Slave)”; moved LL/CON/MAS/BV-53-C, 54-C to new mapping LL 1/5 AND LL 9/9 and feature “PHY Update Procedure (LE Coded PHY, Data Length Extension, Master).</p> <p>TSE 10382 (rating 1): Made editorial changes to test purpose in LL/CON/SLA/BV-49-C, 50-C, 52-C, 53-C, 55-C to 59-C.</p> <p>TSE 10405 (rating 4): Added references 11 and 12 to section 2.1 (References). Added new section 4.9 (Data Flow). Added test cases LL/DFL/MAS/BV-01-C, 02-C, LL/DFL/SLA/BV-01-C, and 02-C and their corresponding TCMT entries.</p>



Revision History	Date	Comments
		<p>TSE 10362 (rating 3): Modified test procedure step 11 and expected outcome inconclusive verdict for LL/CON/SLA/BV-52-C and 53-C.</p> <p>TSE 10288 (rating 3): Replaced MSC with "(TODO)" and modified figure title; test procedure steps 9, 12, 17, and 18; Table 4.20; and expected outcome pass verdict for LL/DDI/SCN/BV-21-C.</p> <p>TSE 9884 (rating 3): Revised MSC and test procedure for LL/SEC/MAS/BI-07-C and 09-C. Revised expected outcome pass verdict for LL/SEC/MAS/BI-09-C.</p> <p>Issue 10377: Replaced the figure and updated the Test Procedure and Pass Verdict for test cases LL/CON/SLA/BV-64-C and LL/CON/MAS/BV-60-C.</p> <p>TSE 9744 (rating 3): Inserted a new step 2 in the LL/TIM/ADV/BV-03-C and 04-C test procedure and renumbered the remaining steps. Corrected the referenced step number in the Pass Verdict from 3 to 4. Expanded the MSC to show the HCI LE Set Extended Scan Response Data command exchange on IUT.</p> <p>TSE 10354 (rating 1): Moved and renumbered LL/DDI/ADV/BV-38-C to LL/DDI/SCN/BV-37-C and revised its corresponding TCMT entry (this was taken care of by TSE 10209). LL/DDI/ADV/BV-38-C is no longer used in the TS.</p> <p>TSE 10357 (rating 1): Globally replaced "Periodic Scanning" with "Scanning for Periodic Advertising" in the TCMT.</p> <p>TSE 10517 (rating 2): Changed 4/7 to 4/6a in the TCMT for test case LL/DDI/SCN/BV-34-C.</p> <p>TSE 10513 (rating 2): Changed 4/8 to 4/7 in the TCMT for test case LL/CON/INI/BV-13-C.</p> <p>TSE 10503 (rating 2): In the TCMT, changed the group containing LL/CON/ADV/BV-01-C to LL 3/2 AND LL 3/7 AND LL 1/4.</p> <p>Changed the group containing LL/CON/ADV/BV-07-C to LL 3/2 AND LL 3/7 AND LL 1/4 AND LL 9/10.</p> <p>TSE 10499 (rating 3): Changed filter value to 0x01 in the Notes for test case LL/DDI/SCN/BI-01-C. Changed specific device to all unknown devices in the Notes for test case LL/DDI/SCN/BI-02-C, and added "OR All White Listed (policy for scanner)" to the Initial Condition State.</p> <p>TSE 10478 (rating 3): Simplified the test purpose for test case LL/CON/SLA/BI-09-C. Replaced the figure with a new figure that includes ALT1 and ALT2. Changed steps 5 and 6 in the test procedure to Alternative 1 and added Alternative 2. Added Alternative 1 and 2 to the Pass Verdict.</p>



Revision History	Date	Comments
		<p>TSE 10353 (rating 3): Added Command Complete events to steps 1, 2, 3, and 10 in the test procedure for test case LL/DDI/SCN/BV-34-C. Changed HCI_Command_Status to HCI_Command_Complete in steps 14 and 17 and in the MSC.</p> <p>TSE 10343 (rating 3): Inserted a new step 1 in the LL/DDI/SCN/BV-33-C test procedure and renumbered the remaining steps. Added HCI_LE_Add_Device_To_White_List to beginning of MSC with complementary HCI Command Complete event.</p> <p>TSE 10016 (rating 3): Editorial fix for step 3 in the LL/CON/MAS/BV-41-C test procedure as it was starting at the wrong point. Added clarifying text to the LL/CON/MAS/BV-42-C test procedure.</p> <p>TSE 10371 (rating 3): Updated step 2 of the LL/CON/SLA/BV-40-C test procedure.</p> <p>TSE 9884 (rating 3): reincorporated, removed step 5 of the LL/SEC/MAS/BI-09-C test procedure.</p> <p>TSE 9975 (rating 3): reincorporated, added Inconclusive Verdict to test case LL/CON/MAS/BV-21-C.</p> <p>TSE 9991 (rating 2): reincorporated, added item LL 9/6 to test cases LL/CON/MAS/BV-53-C and LL/CON/MAS/BV-54-C in the TCMT.</p> <p>TSE 10288 (rating 3): reincorporated, fixed typo in steps 9 and 12 of the LL/DDI/SCN/BV-21-C test procedure.</p> <p>TSE 10610 (rating 3): Revised ADV_EXT_IND and AUX_ADV_IND parameters and changed HCI_LE_SET_Random_Private_Address_Timeout to HCI_LE_SET_Resolvable_Private_Address_Timeout in the MSC for test case LL/DDI/SCN/BV-33-C. Updated step 5 of the Test Procedure.</p> <p>TSE 10371 (rating 3): Updated the MSC to make the LE_PHY_Update_Complete optional for test case LL/CON/SLA/BV-40-C. Revised test procedure steps 2 and 9. Changed step 4 to body text and renumbered the subsequent steps. Revised the pass verdict.</p> <p>TSE 9908 (rating 3): Updated MSC and test procedure steps 1 and 2 for test cases LL/PAC/SLA/BV-01-C and LL/PAC/MAS/BV-01-C to clarify the kind of control PDUs that are supposed to be responded with LL_UNKNOWN_RSP.</p> <p>TSE 10506 (rating 4): Added new test cases LL/CON/ADV/BV-11-C and LL/CON/INI/BV-22-C, and added them to the TCMT.</p>



Revision History	Date	Comments
		<p>TSE 10509 (rating 3): Revised MSC for test case LL/CON/SLA/BV-04-C and updated and renumbered test procedure steps.</p> <p>TSE 10499 (rating 3): Replaced MSC for test case LL/DDI/SCN/BI-02-C (changed HCI_LE_Set_Scan_Parameters command's policy from 0x00 to 0x01).</p> <p>TSE 10530 (rating 3): Replaced MSC to show 40 iterations for test cases LL/ENC/ADV/BI-01-C and 02-C; LL/ENC/SCN/BI-01-C and 02-C; LL/ENC/INI/BI-01-C; and LL/ENC/MAS/BI-01-C. Updated test procedure steps to clarify that the corrupted bit position is changing with each iteration. Doing 2 sequences of 20 iterations makes 40 total. Updated pass verdict and Notes.</p> <p>TSE 10603 (rating 4): Added missing MSC and removed test procedure step 10 for test case LL/CON/MAS/BV-73-C. Added missing MSC and revised test procedure step 4 for test case LL/CON/MAS/BV-74-C. Added missing MSC for test cases LL/CON/MAS/BV-75-C to 80-C.</p> <p>TSE 10389 (rating 3): Modified Round 2 and 3, added new Round 4 in the test procedures for test cases LL/CON/MAS/BV-57-C and LL/CON/SLA/BV-61-C.</p> <p>Fixed integration errors for TSE 10506, 10371, and 10405.</p> <p>Incorporated Core E10734 Pairing Updates TS CR: Updated pass verdict for test cases LL/CON/MAS/BV-13-C and 23-C; LL/CON/SLA/BV-14-C and 22-C.</p> <p>TSE 10800 (rating 1): Corrected test step number references through test procedure in LL/DDI/ADV/BV-24-C.</p>
5.0.3	2018-07-02	Approved by BTI. Prepared for TCRL 2018-1 publication.
5.0.4r00-r20	2018-07-20 – 2018-11-20	<p>Incorporated Core_PAST_CLE_TEST_CR_r05: Added 67 new test cases to TS and TCMT: LL/DDI/SCN/BV-38-C, LL/CON/SLA/BV-88-C – 120-C, LL/CON/MAS/BV-84-C – 116-C. Modified TCMT mapping for LL/DDI/SCN/BV-21, 25-C, 37-C.</p> <p>Incorporated Core_Advertising_Channel_Index_TEST_CR_r04: Modified Test Purpose, Test Procedure, Expected Outcome, Notes for LL/DDI/ADV/BV-01-C – 04-C, 15-C, 16-C, 19-C – 22-C. NOTE: Incorporation of this CR included approved changes from TSE 10559.</p> <p>E10627: Deleted test cases LL/CON/SLA/BV-63-C and LL/CON/MAS/BV-59-C. Added new test case [Constant Tone Extension Request Procedure, IUT Initiated, Unsupported].</p> <p>E10837: Modified test procedure in test cases LL/DDI/SCN/BV-31-C and 32-C.</p> <p>E10699: Deleted test cases LL/CON/SLA/BV-70-C and LL/CON/MAS/BV-66-C. Added new test case [Unrequested Constant Tone Extension, IUT Receiving, AoD].</p>



Revision History	Date	Comments
		<p>E10652: Modified initial condition, test procedure steps 4-5, and pass verdict for test cases LL/DDI/ADV/BV-36-C and 37-C.</p> <p>Incorporated Core Minor Enhancements Batch 1 Test CRr10-clean: Added 17 new test cases: LL/DDI/ADV/BI-03-C – 04-C; LL/DDI/ADV/BV-40-C – 42-C, LL/DDI/ADV/BV-43-C - 44-C, LL/TIM/SLA/BV-08-C – 10-C, LL/TIM/MAS/BV-03-C – 07-C, LL/DDI/ADV/BV-45-C, LL/CON/INI/BV-23-C. Updated test case LL/DDI/ADV/BV-25-C: Modified section heading, Test Procedure step 10, and Pass Verdict bullet 10; added Test Case Configuration section (including new test case LL/DDI/ADV/BV-45-C). Updated test case LL/CON/INI/BV-01-C: Modified section heading, Test Purpose, and steps 2 and 3; added Test Case Configuration section (including new test case LL/CON/INI/BV-23-C). Updated TCMT: Added 12 new test cases LL/TIM/SLA/BV-08-C – 10-C, LL/TIM/MAS/BV-03-C – 07-C, LL/DDI/ADV/BV-43-C - 44-C, LL/DDI/ADV/BV-45-C, LL/CON/INI/BV-23-C; modified mapping for LL/CON/INI/BV-01-C; modified item and feature for LL/DDI/ADV/BV-25-C, LL/TIM/ADV/BV-03-C – 04-C (Note: No feature given in MFE CR for LL/DDI/ADV/BV-45-C). Added 1 new pass verdict each to LL/DDI/SCN/BV-21-C and 24-C. NOTE: Incorporation of this CR included approved changes from TSEs 10579 and 10584.</p> <p>Issue 10708: Updated Extended Advertising, Scannable section: Deleted reference from top of section (references are now in Test Case Configuration table below); modified tenth Pass Verdict; modified references in Test Case Configuration table; TCMT updates: modified mapping for LL/DDI/ADV/BV-25-C, LL/DDI/ADV/BV-45-C, LL/TIM/ADV/BV-03-C – 04-C; added missing feature for LL/DDI/ADV/BV-45-C.</p> <p>Issue 10709: Updated Extended Advertising, Scannable section: Global. Changed “Empty” to “Not Present” in Test Case Configuration table.</p> <p>Issue 10711: Modified Initial Condition, Test Procedure, and/or Pass Verdict for LL/TIM/SLA/BV-08-C – 10-C, LL/TIM/MAS/BV-03-C – 07-C.</p> <p>Issue 10712: Modified Test Purpose for LL/DDI/ADV BI-03-C – 04-C.</p> <p>Issue 10716: Reverted all changes in MFE CR to LL/CON/INI/BV-01-C, in spec text and TCMT.</p> <p>Issue 10781: Modified Test Procedure and Pass Verdict for LL/DDI/ADV/BV-43-C.</p> <p>Issue 10826: Added 2 new test cases to spec text. LL/DDI/ADV/BI-05-C - 06-C. TCMT entries appear in HCI.TS TCMT.</p> <p>E10745: Deleted test cases LL/CON/SLA/BV-60-C to 62-C, 64-C to 69-C, and 71-C to 76-C; LL/CON/MAS/BV-56-C to 58-C, 60-C to 65-C, and 67-C to 68-C. Added new test cases:</p>



Revision History	Date	Comments
		<p>[Constant Tone Extension Request Procedure, IUT Initiated, AoA]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, Periodic]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, Timeout]</p> <p>[Constant Tone Extension Request Procedure, IUT Responding, AoA]</p> <p>[Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, AoD]</p> <p>[Constant Tone Extension Request Procedure, IUT Responding, AoD]</p> <p>[Unrequested Constant Tone Extension, IUT Receiving, AoA]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection]</p> <p>[Constant Tone Extension Request Procedure, IUT Responding, AoA, Encrypted Connection]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection]</p> <p>[Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC]</p> <p>[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC]</p> <p>Issue 11105: In LL/DDI/ADV/BV-43-C, replaced AUX_SYNC_IND by "periodic advertising train"; made additional related clarifications to the test procedure.</p> <p>Issue 10859: LL/TIM/SLA/BV-08-C, 10-C; LL/TIM/MAS/BV-03-c – 07-C: Made corrections to test cases to allow Sleep Clock Accuracy procedure to trigger. Removed last paragraph of Initial Condition. Replaced MSC. Edited test procedure steps.</p> <p>Issue 11130: Modified steps 4 and 5 and added a fail verdict to LL/CON/SLA/BV-106-C and LL/CON/MAS/BV-102-C.</p> <p>Issue 11122: Deleted test cases belonging to enhancements out of the final Madrid scope: DDI/ADV/BI-03-C and 04-C; DDI/ADV/BV-40-C through 42-C; DDI/ADV/BV-44-C. Updated test case mapping accordingly.</p> <p>TSE 9909 (rating 3): Updated MSC and test procedure step 3 for test case LL/CON/SLA/BV-13-C.</p> <p>TSE 10021 (rating 4): Added new test case LL/CON/MAS/BV-117-C, and updated TCMT and section 4.1.5.8</p>



Revision History	Date	Comments
		<p>Pass/Inconclusive/Fail Verdict Conventions Table 4.2: Test cases with allowable Inconclusive Verdicts with new test case.</p> <p>TSE 10345 (rating 2): Updated section title, test procedure, MSC, expected outcome, and pass verdict for test case LL/DDI/ADV/BV-24-C. Updated TCMT with test cases LL/DDI/ADV/BV-TBD1-C and TBD2-C.</p> <p>TSE 10649 (rating 3): Updated initial condition for test case LL/DDI/SCN/BV-33-C.</p> <p>TSE 10190 (rating 3): Updated test procedure steps and fixed punctuation in pass verdict for test case LL/DDI/ADV/BV-09-C.</p> <p>TSE 10668 (rating 1): Updated test procedure step 11 and pass verdict for test case LL/DDI/ADV/BV-25-C [Extended Advertising, Scannable].</p> <p>TSE 10675 (rating 1): Removed "non white" listed device address from test procedure steps for test cases LL/DDI/ADV/BV-05-C, 06-C, and 17-C; LL/DDI/SCN/BV-02-C and 04-C; LL/CON/INI/BV-06-C and 07-C.</p> <p>TSE 10697 (rating 3): Added a note at the end of the test procedure for test case LL/DDI/SCN/BV-19-C.</p> <p>TSE 10702 (rating 1): Changed "Upper Tester" to "Lower Tester" in test procedure step 4 for test case LL/DDI/ADV/BV-06-C.</p> <p>TSE 10713 (rating 1): Updated test procedure step 11 for test case LL/CON/MAS/BV-74-C.</p> <p>TSE 10768 (rating 1): Updated MSC for test case LL/CON/SLA/BV-26-C.</p> <p>TSE 10807 (rating 3): Added optional sequence of test steps "LE PHY Update Procedure (LE Coded Switch) Optional Test Steps" to the end of section 4.1.5.7.3 Optional Test Steps.</p> <p>TSE 10809 (rating 1): Revised test procedure step 14 for test case LL/DDI/ADV/BV-05-C.</p> <p>TSE 10812 (rating 3): Updated test procedure steps 10 and 13 for test case LL/DDI/SCN/BV-33-C.</p> <p>TSE 10817 (rating 3): Updated test procedure and pass verdict for test case LL/SEC/MAS/BV-10-C.</p> <p>TSE 10866 (rating 3): Updated test procedure step 1 for test case LL/CON/SLA/BV-18-C.</p> <p>TSE 10900 (rating 1): Updated reference for test cases LL/ENC/ADV/BI-01-C and 02-C, LL/ENC/SCN/BI-01-C and 02-C, LL/ENC/INI/BI-01-C, LL/ENC/SLA/BI-01-C, and LL/ENC/MAS/BI-01-C.</p> <p>TSE 10989 (rating 1): Updated MSCs for test case LL/DDI/ADV/BV-30-C.</p> <p>TSE 11050 (rating 1): Updated MSCs for test cases LL/DFL/SLA/BV-01-C and 02-C; and LL/DFL/MAS/BV-01-C and 02-C.</p>



Revision History	Date	Comments
		<p>TSE 11088 (rating 3): Updated test procedure step 6 for test case LL/DDI/ADV/BV-32-C.</p> <p>TSE 11101 (rating 1): Updated MSC and test procedure step 9 for test case LL/CON/MAS/BV-50-C.</p> <p>TSE 10190 (rating 3): Updated test procedure steps per new CR for test case LL/DDI/ADV/BV-09-C.</p> <p>TSE 10656 (rating 2): Updated test procedure for test case LL/CON/MAS/BV-18-C.</p> <p>TSE 10808 (rating 3): Updated MSCs, test procedure, and expected outcome for test cases LL/CON/SLA/BV-57-C and LL/CON/MAS/BV-55-C.</p> <p>TSE 11104 (rating 4): Added new test case LL/SEC/ADV/BV-21-C.</p> <p>TSE 11141 (rating 4): Added new test cases LL/CON/INI/BV-23-C and 24-C.</p> <p>Issue 10826: Incorporation error corrections to LL/DDI/ADV/BV-06-C: Replaced MSC diagram and modified MSC title, Test Procedure steps 1, 2, and 4, and Pass Verdict.</p> <p>TSE 10345 (rating 2): Replaced all instances of LL/DDI/ADV/BV-TBD1-C with 48-C and LL/DDI/ADV/BV-TBD2-C with 49-C.</p> <p>TSE 10809 (rating 1): Revised test procedure step 14 (removed the words "steps 3" but not the rest of the sentence) for test case LL/DDI/ADV/BV-05-C.</p> <p>TSE 11153 (rating 1): Removed Table 4.2 (Test cases with allowable Inconclusive verdicts).</p> <p>TSE 11104 (rating 4): Updated MSC and TCMT entry for test case LL/SEC/ADV/BV-21-C.</p> <p>TSE 11141 (rating 4): Updated MSCs and TCMT entries for test cases LL/CON/INI/BV-23-C and 24-C.</p> <p>Fixed the remaining CONN_REQ / CONNECT_REQ instances by replacing with CONNECT_IND.</p> <p>TSE 10584 (rating 3): Added inconclusive verdict to LL/DDI/SCN/BV-24-C</p> <p>Updated TC names following BTI decision on 11/05/18</p> <p>TSE 10584 (rating 3): Removed text from pass verdict LL/DDI/SCN/BV-24-C</p> <p>Fixed the remaining CONN_REQ / CONNECT_REQ instances by replacing with CONNECT_IND. Corrected step numbering per comment 43185 om issue 10627.</p> <p>Production edits: Updated all TBD (x) cross-reference values. For all new Madrid (5.1) text, changed style from gray to black.</p>
5.1.0	2018-11-13	Updated revision number from 5.0.4 to 5.1.0 to align with the adoption of Core Specification version 5.1
5.1.0	2018-12-07	Approved by BTI. Prepared for TCRL 2018-2 publication.



Contributors

Name	Company
Angel Romero	AT4 wireless
Antonio Narváez Castillo	AT4 wireless
Elisa Rincón	AT4 wireless
Francisco Narvaez	AT4 wireless
Juan Manuel Hidalgo	AT4 wireless
Angel Polo	Broadcom
Norbert Grunert	Broadcom
Giriraj Goyal	CSR
Joe Decuir	CSR
Magnus Sommansson	CSR
Mayank Batra	CSR
Neil Stewart	CSR
Robin Heydon	CSR
Harish Balasubramaniam	Intel
Magnus Eriksson	Intel
Oren Haggai	Intel
Yao Wang	IVT Corporation
Rajan S Pallathu	MindTree Ltd.
Jukka Reunamäki	Nokia
Mika Kasslin	Nokia
David Lopes	Nordic Semiconductor
Joel Linsky	Qualcomm Atheros
Jean-Philippe Lambert	RivieraWaves
Rasmus Abildgren	Samsung Electronics
Paul Vanoostende	STMicroelectronics
Rogier Scheffer	STMicroelectronics
Tony Viscardi	Texas Instruments
Martti Söderlund	TietoEnator
Jason Hillyard	Wicentric
David Engelien-Lopes	Nordic Semiconductor ASA
Angel Polo	Broadcom
Brian A. Redding	Qualcomm
Rasmus Abildgren	Samsung Electronics Co., Ltd
Mayank Batra	CSR
Harish Balasubramaniam	Intel



Use of this specification is your acknowledgement that you agree to and will comply with the following notices and disclaimers. You are advised to seek appropriate legal, engineering, and other professional advice regarding the use, interpretation, and effect of this specification.

Use of Bluetooth specifications by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Any use of this specification by a member that is not in compliance with the applicable membership and other related agreements is prohibited and, among other things, may result in (i) termination of the applicable agreements and (ii) liability for infringement of the intellectual property rights of Bluetooth SIG and its members.

Use of this specification by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members. The furnishing of this specification does not grant any license to any intellectual property of Bluetooth SIG or its members. THIS SPECIFICATION IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR THAT THE CONTENT OF THIS SPECIFICATION IS FREE OF ERRORS. For the avoidance of doubt, Bluetooth SIG has not made any search or investigation as to third parties that may claim rights in or to any specifications or any intellectual property that may be required to implement any specifications and it disclaims any obligation or duty to do so.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS SPECIFICATION AND ANY INFORMATION CONTAINED IN THIS SPECIFICATION, INCLUDING LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF THE DAMAGES.

If this specification is a prototyping specification, it is solely for the purpose of developing and using prototypes to verify the prototyping specifications at Bluetooth SIG sponsored IOP events. Prototyping Specifications cannot be used to develop products for sale or distribution and prototypes cannot be qualified for distribution.

Products equipped with Bluetooth wireless technology ("Bluetooth Products") and their combination, operation, use, implementation, and distribution may be subject to regulatory controls under the laws and regulations of numerous countries that regulate products that use wireless non-licensed spectrum. Examples include airline regulations, telecommunications regulations, technology transfer controls and health and safety regulations. You are solely responsible for complying with all applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses in connection with your use of this specification and development, manufacture, and distribution of Bluetooth Products. Nothing in this specification provides any information or assistance in connection with complying with applicable laws or regulations or obtaining required authorizations, permits, or licenses.

Bluetooth SIG is not required to adopt any specification or portion thereof. If this specification is not the final version adopted by Bluetooth SIG's Board of Directors, it may not be adopted. Any specification adopted by Bluetooth SIG's Board of Directors may be withdrawn, replaced, or modified at any time. Bluetooth SIG reserves the right to change or alter final specifications in accordance with its membership and operating agreements.

Copyright © 2009–2019. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, Lenovo (Singapore) Pte. Ltd., Microsoft Corporation, Nokia Corporation, and Toshiba Corporation. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.



Contents

1 Scope	49
2 References, Definitions, and Abbreviations	50
2.1 References	50
3 Test Suite Structure (TSS)	51
3.1 Test Strategy	51
3.2 Test Groups	53
3.2.1 Device Discovery	53
3.2.2 Connection Handling.....	53
3.2.3 Timing	53
3.2.4 Radio Frame Encoding	53
3.2.5 Frequency Hopping.....	53
3.2.6 Packet Formats.....	53
3.2.7 Security.....	53
3.3 Behavior Tests.....	53
3.3.1 Valid Behavior (BV) Tests.....	54
3.3.2 Invalid Behavior (BI) Tests	54
3.4 Test Realization	54
4 Test Cases (TC).....	55
4.1 Introduction	55
4.1.1 Test Case Naming Conventions	55
4.1.2 Conformance	55
4.1.3 Common Packet Contents	56
4.1.3.1 Access Address Used	56
4.1.4 Common Message Contents	56
4.1.4.1 Checksums Used	56
4.1.4.2 Device Addresses Used	57
4.1.5 Common Initial and Final Conditions	57
4.1.5.1 ADV	62
4.1.5.2 SCN.....	67
4.1.5.3 INI.....	68
4.1.5.4 SLA.....	70
4.1.5.5 MAS.....	72
4.1.5.6 SEC	74
4.1.5.7 Common Test Procedure Steps	76
4.1.5.8 Pass/Inconclusive/Fail Verdict Conventions	82
4.1.6 Data Fragmentation over HCI	83
4.1.7 Data PDUs and Empty PDUs	83
4.1.8 Outstanding Commands Prior to Disconnection	83
4.2 DDI.....	83
4.2.1 Common PDU Contents	83
4.2.1.1 Legacy Advertising	84
4.2.1.2 Extended Advertising.....	85
4.2.2 ADV	85
4.2.2.1 Common PDU Contents	85
4.2.2.2 LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events].....	85
4.2.2.3 LL/DDI/ADV/BV-02-C [Undirected Advertising Events]	88



4.2.2.4	LL/DDI/ADV/BV-03-C [Advertising Data: Non-Connectable]	91
4.2.2.5	LL/DDI/ADV/BV-04-C [Advertising Data: Undirected]	93
4.2.2.6	LL/DDI/ADV/BV-05-C [Scan Request: Undirected Connectable]	96
4.2.2.7	LL/DDI/ADV/BV-06-C [Connection Request]	98
4.2.2.8	LL/DDI/ADV/BV-07-C [Scan Request Connection Request]	100
4.2.2.9	LL/DDI/ADV/BV-08-C [Scan Request Device Filtering]	102
4.2.2.10	LL/DDI/ADV/BV-09-C [Connection Request Device Filtering]	105
4.2.2.11	LL/DDI/ADV/BV-11-C [Directed Advertising Events]	108
4.2.2.12	LL/DDI/ADV/BV-15-C [Discoverable Advertising Events]	111
4.2.2.13	LL/DDI/ADV/BV-16-C [Advertising Data: Discoverable]	113
4.2.2.14	LL/DDI/ADV/BV-17-C [Scan Request: Discoverable]	115
4.2.2.15	LL/DDI/ADV/BV-18-C [Device Filtering: Discoverable]	117
4.2.2.16	LL/DDI/ADV/BI-01-C [Scan Request Invalid CRC]	120
4.2.2.17	LL/DDI/ADV/BI-02-C [Connection Request Invalid CRC]	122
4.2.2.18	LL/DDI/ADV/BV-19-C [Low Duty Cycle Directed Advertising Events]	123
4.2.2.19	LL/DDI/ADV/BV-20-C [Advertising Always Using the LE 1M PHY]	126
4.2.2.20	LL/DDI/ADV/BV-21-C [Extended Advertising, Legacy PDUs, Non-Connectable]	128
4.2.2.21	LL/DDI/ADV/BV-22-C [Extended Advertising, Legacy PDUs, Undirected]	130
4.2.2.22	Extended Advertising, Non-Connectable	132
4.2.2.23	[Extended Advertising, Scannable]	137
4.2.2.24	LL/DDI/ADV/BV-26-C [Extended Advertising, Periodic Advertising]	142
4.2.2.25	LL/DDI/ADV/BV-27-C [Extended Advertising, Host Modifying Data and ADI]	146
4.2.2.26	LL/DDI/ADV/BV-28-C [Extended Advertising, Overlapping Extended Advertising Events]	150
4.2.2.27	LL/DDI/ADV/BV-29-C [Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY]	152
4.2.2.28	LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY]	156
4.2.2.29	LL/DDI/ADV/BV-31-C [Extended Advertising, Multiple Sets, Multiple PHYs, LE 1M and LE Coded PHYs]	160
4.2.2.30	LL/DDI/ADV/BV-32-C [Extended Advertising, Multiple Sets, Legacy and Extended]	164
4.2.2.31	LL/DDI/ADV/BV-33-C [Extended Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)]	168
4.2.2.32	LL/DDI/ADV/BV-34-C [Extended Advertising, TX Power with RF Path Compensation]	172
4.2.2.33	LL/DDI/ADV/BV-35-C [Extended Advertising, Multiple Sets, Maximum Supported Sets]	174
4.2.2.34	LL/DDI/ADV/BV-36-C [AoD Connectionless CTE Advertising]	176
4.2.2.35	LL/DDI/ADV/BV-37-C [AoA Connectionless CTE Advertising]	180
4.2.2.36	LL/DDI/ADV/BV-39-C [Connectionless CTE Advertising – Maintain CTE Configuration]	184
4.2.2.37	LL/DDI/ADV/BV-43-C [Periodic Advertising validating SyncInfo fields]	188
4.2.2.38	LL/DDI/ADV/BI-05-C [Disallow Extended Advertising PDU sizes for Legacy Advertising when advertising enabled]	190
4.2.2.39	LL/DDI/ADV/BI-06-C [Disallow Extended Advertising PDU sizes for Scannable Legacy Advertising when advertising enabled]	191
4.2.3	SCN	192
4.2.3.1	Common PDU Contents	192
4.2.3.2	LL/DDI/SCN/BV-01-C [Passive Scanning: Non-connectable]	193
4.2.3.3	LL/DDI/SCN/BV-02-C [Passive Scanning Device Filtering]	195
4.2.3.4	LL/DDI/SCN/BV-03-C [Active Scanning]	197
4.2.3.5	LL/DDI/SCN/BV-04-C [Active Scanning Device Filtering]	199
4.2.3.6	LL/DDI/SCN/BV-05-C [Scanning For Advertiser Types]	202
4.2.3.7	LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected Events]	205
4.2.3.8	LL/DDI/SCN/BV-11-C [Passive Scanning: Directed Events]	207
4.2.3.9	LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable Events]	209



4.2.3.10	LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]	211
4.2.3.11	LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning: Directed Events to an address different from the scanner's address].....	213
4.2.3.12	LL/DDI/SCN/BV-15-C [Network Privacy – Active Scanning, no Local IRK, no Peer IRK].....	215
4.2.3.13	LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK].....	217
4.2.3.14	LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK].....	219
4.2.3.15	LL/DDI/SCN/BV-18-C [Network Privacy – Active Scanning, Local IRK, Peer IRK].....	221
4.2.3.16	LL/DDI/SCN/BV-19-C [Extended Scanning, Passive].....	223
4.2.3.17	LL/DDI/SCN/BV-20-C [Extended Scanning, Active]	227
4.2.3.18	LL/DDI/SCN/BV-21-C [Extended Scanning, Periodic Advertising Reception]	230
4.2.3.19	LL/DDI/SCN/BV-23-C [Extended Scanning, Multiple Sets, Passive, Multiple PHYs (All Supported PHYs)].....	235
4.2.3.20	LL/DDI/SCN/BV-24-C [Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)].....	238
4.2.3.21	LL/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)].....	241
4.2.3.22	LL/DDI/SCN/BV-26-C [Network Privacy – Passive Scanning, Peer IRK, Ignore Identity Address]	245
4.2.3.23	LL/DDI/SCN/BV-28-C [Device Privacy – Passive Scanning, Peer IRK, Accept Identity Address]	247
4.2.3.24	LL/DDI/SCN/BV-29-C [AoD Connectionless CTE Scanning].....	249
4.2.3.25	LL/DDI/SCN/BV-30-C [AoA Connectionless CTE Scanning].....	253
4.2.3.26	LL/DDI/SCN/BV-31-C [AoD Connectionless CTE Scanning, Incorrect CRC]	257
4.2.3.27	LL/DDI/SCN/BV-32-C [AoA Connectionless CTE Scanning, Incorrect CRC]	257
4.2.3.28	LL/DDI/SCN/BV-33-C [Privacy - Extended Scanning, Active]	258
4.2.3.29	LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]	261
4.2.3.30	LL/DDI/SCN/BV-35-C [Connectionless CTE Scanning, Filter Wrong CTE Types on Synchronization].....	265
4.2.3.31	LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]	268
4.2.3.32	LL/DDI/SCN/BV-37-C [First AUX_SYNC_IND Never Received]	271
4.2.3.33	LL/DDI/SCN/BV-38-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled].....	273
4.2.3.34	LL/DDI/SCN/BI-01-C [Active Scanning Invalid CRC].....	276
4.2.3.35	LL/DDI/SCN/BI-02-C [Passive Scanning Invalid CRC]	278
4.2.3.36	LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Wrong AdvA in Response]	280
4.3	CON	282
4.3.1	Common PDU Contents	282
4.3.2	ADV	283
4.3.2.1	Common PDU Contents	284
4.3.2.2	Common Variables	284
4.3.2.3	LL/CON/ADV/BV-01-C [Accepting Connections].....	284
4.3.2.4	LL/CON/ADV/BV-02-C [Accepting Connections Timeout]	286
4.3.2.5	LL/CON/ADV/BV-03-C [Master Missing Slave Packets].....	288
4.3.2.6	LL/CON/ADV/BV-04-C [Directed Advertising Connection]	290
4.3.2.7	LL/CON/ADV/BV-05-C [Extended Advertising, Accepting Connections]	292
4.3.2.8	LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections]	296
4.3.2.9	LL/CON/ADV/BV-07-C [Accepting Connections, Channel Selection Algorithm #1]	299
4.3.2.10	LL/CON/ADV/BV-08-C [Directed Advertising Connection, Channel Selection Algorithm #1]	301
4.3.2.11	LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2].....	303
4.3.2.12	LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2]	305
4.3.2.13	LL/CON/ADV/BI-01-C [Connection Supervision Timeout during Fail Connection Setup]	307



4.3.2.14	LL/CON/ADV/BV-11-C [Accepting Connections, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2].....	309
4.3.3	INI	311
4.3.3.1	LL/CON/INI/BV-01-C [Connection Initiation].....	312
4.3.3.2	LL/CON/INI/BV-02-C [Connecting to Directed Advertising]	316
4.3.3.3	LL/CON/INI/BV-03-C [Connection Initiation Missed Replies].....	318
4.3.3.4	LL/CON/INI/BV-04-C [Connection Initiation Timeout]	320
4.3.3.5	LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected]	322
4.3.3.6	LL/CON/INI/BV-07-C [Initiation Device Filtering: Directed]	324
4.3.3.7	LL/CON/INI/BV-08-C [Network Privacy – Connection Establishment responding to connectable undirected advertising, Initiator]	326
4.3.3.8	LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator].....	328
4.3.3.9	LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator].....	330
4.3.3.10	LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator]	332
4.3.3.11	LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator]	335
4.3.3.12	LL/CON/INI/BV-13-C [Extended Scanning, Connection Initiation]	337
4.3.3.13	LL/CON/INI/BV-14-C [Connection Initiation, Channel Selection Algorithm #1].....	341
4.3.3.14	LL/CON/INI/BV-15-C [Connecting to Directed Advertising, Channel Selection Algorithm #1]	344
4.3.3.15	LL/CON/INI/BV-16-C [Connection Initiation, Channel Selection Algorithm #2].....	347
4.3.3.16	LL/CON/INI/BV-17-C [Connecting to Directed Advertising, Channel Selection Algorithm #2]	350
4.3.3.17	LL/CON/INI/BV-18-C [Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address]	353
4.3.3.18	LL/CON/INI/BV-19-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Ignore Identity Address]	354
4.3.3.19	LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address]	356
4.3.3.20	LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Accept Identity Address]	358
4.3.3.21	LL/CON/INI/BI-01-C [Connection Initiation Invalid CRC]	360
4.3.3.22	LL/CON/INI/BI-02-C [Slave Packets Invalid CRC].....	361
4.3.3.23	LL/CON/INI/BV-22-C [Connection Initiation, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2]	364
4.3.3.24	LL/CON/INI/BV-23-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled].....	367
4.3.3.25	LL/CON/INI/BV-24-C [Network Privacy - Connection Establishment using resolving list with address resolution disabled].....	368
4.3.4	SLA	370
4.3.4.1	LL/CON/SLA/BV-02-C [Slave Asymmetric Connections].....	370
4.3.4.2	LL/CON/SLA/BV-04-C [Slave Sending Data].....	372
4.3.4.3	LL/CON/SLA/BV-05-C [Slave Receiving Data]	375
4.3.4.4	LL/CON/SLA/BV-06-C [Slave Sending and Receiving Data].....	377
4.3.4.5	LL/CON/SLA/BV-10-C [Accepting Parameter Update]	379
4.3.4.6	LL/CON/SLA/BV-11-C [Slave Sending Termination]	381
4.3.4.7	LL/CON/SLA/BV-12-C [Slave Accepting Termination].....	383
4.3.4.8	LL/CON/SLA/BV-13-C [Slave Supervision Timer]	384
4.3.4.9	LL/CON/SLA/BV-14-C [Feature Setup Response]	385
4.3.4.10	LL/CON/SLA/BV-15-C [Slave Retransmission Request]	386



4.3.4.11	LL/CON/SLA/BV-16-C [Slave Retransmission].....	387
4.3.4.12	LL/CON/SLA/BV-17-C [Slave Acknowledgement Repetition].....	389
4.3.4.13	LL/CON/SLA/BV-18-C [Slave Lost Negative Acknowledgement]	390
4.3.4.14	LL/CON/SLA/BV-19-C [Slave Request Version].....	391
4.3.4.15	LL/CON/SLA/BV-20-C [Slave Respond Version].....	393
4.3.4.16	LL/CON/SLA/BV-21-C [Slave Acknowledgement Scheme].....	393
4.3.4.17	LL/CON/SLA/BV-22-C [Initiate Feature Exchange]	395
4.3.4.18	LL/CON/SLA/BV-23-C [Initiate Feature Exchange – Master does not support].....	396
4.3.4.19	LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept].....	397
4.3.4.20	LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject].....	401
4.3.4.21	LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision].....	402
4.3.4.22	LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]	405
4.3.4.23	LL/CON/SLA/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]	407
4.3.4.24	LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity]	409
4.3.4.25	LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only]	413
4.3.4.26	LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity]	416
4.3.4.27	LL/CON/SLA/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points].....	417
4.3.4.28	LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked]	419
4.3.4.29	LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects].....	420
4.3.4.30	LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure]	421
4.3.4.31	LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure].....	426
4.3.4.32	LL/CON/SLA/BV-43-C [Responding to PHY Update Procedure – Symmetric Only].....	431
4.3.4.33	LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure].....	432
4.3.4.34	LL/CON/SLA/BV-45-C [Protocol Timeout for PHY Update Procedure].....	434
4.3.4.35	LL/CON/SLA/BV-46-C [Handling Protocol Collision – Different Procedure – Channel Map]	436
4.3.4.36	LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters].....	438
4.3.4.37	LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update].....	440
4.3.4.38	LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]	442
4.3.4.39	LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions].....	446
4.3.4.40	LL/CON/SLA/BV-51-C [Protocol Timeout for PHY Update Procedure – No Update Request]	450
4.3.4.41	LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change].....	452
4.3.4.42	LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change].....	455
4.3.4.43	LL/CON/SLA/BV-54-C [Slave Receiving Data, LE Coded, CI Change]	460
4.3.4.44	LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded].....	463
4.3.4.45	LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded].....	467
4.3.4.46	LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded]	471
4.3.4.47	LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]	475



4.3.4.48	LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded].....	478
4.3.4.49	LL/CON/SLA/BV-77-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]	481
4.3.4.50	LL/CON/SLA/BV-78-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]	481
4.3.4.51	LL/CON/SLA/BV-79-C [Slave Data Length Update – Master does not support; LE 1M PHY].....	481
4.3.4.52	LL/CON/SLA/BV-80-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]	481
4.3.4.53	LL/CON/SLA/BV-81-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]	482
4.3.4.54	LL/CON/SLA/BV-82-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY].....	482
4.3.4.55	LL/CON/SLA/BV-83-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY].....	482
4.3.4.56	LL/CON/SLA/BV-84-C [Slave Data Length Update – Master does not support; LE Coded PHY].....	482
4.3.4.57	LL/CON/SLA/BV-85-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]	482
4.3.4.58	LL/CON/SLA/BV-86-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]	484
4.3.4.59	LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]	485
4.3.4.60	LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]	487
4.3.4.61	LL/CON/SLA/BI-04-C [Rejecting Connection Change]	488
4.3.4.62	LL/CON/SLA/BI-05-C [Slave Connection Control Timer].....	490
4.3.4.63	LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]	492
4.3.4.64	LL/CON/SLA/BI-08-C [Accepting Connection Parameter Request – Illegal Parameters]	493
4.3.4.65	LL/CON/SLA/BI-09-C [Responding to PHY Update Procedure – Instant In Past]	494
4.3.5	MAS	496
4.3.5.1	LL/CON/MAS/BV-02-C [Events with Slave Latency]	496
4.3.5.2	LL/CON/MAS/BV-03-C [Master Sending Data].....	498
4.3.5.3	LL/CON/MAS/BV-04-C [Master Receiving Data].....	501
4.3.5.4	LL/CON/MAS/BV-05-C [Master Sending and Receiving Data].....	503
4.3.5.5	LL/CON/MAS/BV-07-C [Requesting Parameter Update].....	504
4.3.5.6	LL/CON/MAS/BV-08-C [Master Sending Termination]	506
4.3.5.7	LL/CON/MAS/BV-09-C [Master Accepting Termination].....	507
4.3.5.8	LL/CON/MAS/BV-10-C [Master Supervision Timer]	508
4.3.5.9	LL/CON/MAS/BV-13-C [Feature Setup Request]	510
4.3.5.10	LL/CON/MAS/BV-14-C [Master Retransmission Request]	511
4.3.5.11	LL/CON/MAS/BV-15-C [Master Retransmission].....	512
4.3.5.12	LL/CON/MAS/BV-16-C [Master Acknowledgement Repetition]	514
4.3.5.13	LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]	515
4.3.5.14	LL/CON/MAS/BV-18-C [Master Latency Retransmission Request].....	517
4.3.5.15	LL/CON/MAS/BV-19-C [Connection Control Timeout].....	519
4.3.5.16	LL/CON/MAS/BV-20-C [Master Request Version].....	520
4.3.5.17	LL/CON/MAS/BV-21-C [Master Respond Version].....	521
4.3.5.18	LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]	522
4.3.5.19	LL/CON/MAS/BV-23-C [Responding to Feature Exchange]	524
4.3.5.20	LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]	525
4.3.5.21	LL/CON/MAS/BV-25-C [Initiating Connection Parameter Request – Reject].....	528



4.3.5.22	LL/CON/MAS/BV-26-C [Initiating Connection Parameter Request – same procedure collision].....	530
4.3.5.23	LL/CON/MAS/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]	532
4.3.5.24	LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]	534
4.3.5.25	LL/CON/MAS/BV-29-C [Initiating Connection Parameter Request – remote legacy host].....	536
4.3.5.26	LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred_Periodicity]	538
4.3.5.27	LL/CON/MAS/BV-31-C [Accepting Connection Parameter Request – preferred anchor points only]	541
4.3.5.28	LL/CON/MAS/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity]	544
4.3.5.29	LL/CON/MAS/BV-33-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points].....	545
4.3.5.30	LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked]	547
4.3.5.31	LL/CON/MAS/BV-35-C [Accepting Connection Parameter Request – Host rejects].....	548
4.3.5.32	LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]	549
4.3.5.33	LL/CON/MAS/BV-42-C [Initiating PHY Update Procedure – Symmetric Only]	553
4.3.5.34	LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure].....	554
4.3.5.35	LL/CON/MAS/BV-44-C [Responding to PHY Update Procedure – Symmetric Only].....	559
4.3.5.36	LL/CON/MAS/BV-45-C [Handling Protocol Collision – Same Procedure].....	559
4.3.5.37	LL/CON/MAS/BV-46-C [Protocol Timeout for PHY Update Procedure].....	561
4.3.5.38	LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]	563
4.3.5.39	LL/CON/MAS/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Parameters].....	565
4.3.5.40	LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]	567
4.3.5.41	LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions].....	571
4.3.5.42	LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]	575
4.3.5.43	LL/CON/MAS/BV-52-C [Master Receiving Data, LE Coded, CI Change]	577
4.3.5.44	LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded].....	580
4.3.5.45	LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded].....	584
4.3.5.46	LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded]	588
4.3.5.47	LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]	591
4.3.5.48	LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]	594
4.3.5.49	LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY]	597
4.3.5.50	LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]	600
4.3.5.51	LL/CON/MAS/BV-77-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]	602
4.3.5.52	LL/CON/MAS/BV-78-C [Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY].....	604
4.3.5.53	LL/CON/MAS/BV-79-C [Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY].....	606
4.3.5.54	LL/CON/MAS/BV-80-C [Master Data Length Update – Slave does not support; LE Coded PHY]	608
4.3.5.55	LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]	610



4.3.5.56	LL/CON/MAS/BV-82-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]	612
4.3.5.57	LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]	614
4.3.5.58	LL/CON/MAS/BI-02-C [Master T_Terminate Timer]	615
4.3.5.59	LL/CON/MAS/BI-04-C [Master Connection Control Timer]	616
4.3.5.60	LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]	618
4.3.5.61	LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]	619
4.3.6	Both Connected Roles	621
4.3.6.1	Constant Tone Extension Request Procedure, IUT Responding, Unsupported	621
4.3.6.2	[Periodic Advertising Sync Transfer Procedure, Advertising IUT Initiated]	623
4.3.6.3	[Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]	627
4.3.6.4	[Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYs]	630
4.3.6.5	[Periodic Advertising Sync Transfer Procedure, Accepting – Skipping Events]	632
4.3.6.6	[Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]	634
4.3.6.7	[Periodic Advertising Sync Transfer Procedure, Accepting – Extreme Timings]	636
4.3.6.8	[Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure]	638
4.3.6.9	[Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses]	640
4.3.6.10	[Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]	645
4.3.6.11	[Acknowledging Long Control PDUs]	646
4.3.6.12	[Rejecting Request to Send Long Control PDUs before Feature Exchange]	648
4.3.6.13	[Rejecting Request to Send Long Control PDUs after Feature Exchange]	650
4.3.6.14	[Sending Long Control PDUs after Feature Exchange]	652
4.3.6.15	[Constant Tone Extension Request Procedure, IUT Initiated, Unsupported]	653
4.3.6.16	[Unrequested Constant Tone Extension, IUT Receiving, AoD]	655
4.3.6.17	[Constant Tone Extension Request Procedure, IUT Initiated, AoA]	657
4.3.6.18	[Constant Tone Extension Request Procedure, IUT Initiated, Periodic]	660
4.3.6.19	[Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]	662
4.3.6.20	[Constant Tone Extension Request Procedure, IUT Initiated, Timeout]	664
4.3.6.21	[Constant Tone Extension Request Procedure, IUT Responding, AoA]	665
4.3.6.22	[Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]	667
4.3.6.23	[Constant Tone Extension Request Procedure, IUT Initiated, AoD]	668
4.3.6.24	[Constant Tone Extension Request Procedure, IUT Responding, AoD]	670
4.3.6.25	[Unrequested Constant Tone Extension, IUT Receiving, AoA]	672
4.3.6.26	[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection]	676
4.3.6.27	[Constant Tone Extension Request Procedure, IUT Responding, AoA, Encrypted Connection]	676
4.3.6.28	[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection]	677
4.3.6.29	[Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]	677
4.3.6.30	[Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC]	678
4.3.6.31	[Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC]	679
4.4	TIM	680
4.4.1	ADV	680
4.4.1.1	LL/TIM/ADV/BV-01-C [Earliest Transmission to Advertiser]	680
4.4.1.2	LL/TIM/ADV/BV-02-C [Latest Transmission to Advertiser]	682
4.4.1.3	LL/TIM/ADV/BV-03-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser]	684



4.4.1.4	LL/TIM/ADV/BV-04-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser]	686
4.4.2	SLA	688
4.4.2.1	LL/TIM/SLA/BV-01-C [Adjusting Anchor Point].....	688
4.4.2.2	LL/TIM/SLA/BV-02-C [Earliest Transmission Start to Slave]	690
4.4.2.3	LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]	692
4.4.2.4	LL/TIM/SLA/BV-04-C [Shortest Connection Interval].....	694
4.4.2.5	LL/TIM/SLA/BV-05-C [Longest Connection Interval]	695
4.4.2.6	LL/TIM/SLA/BV-06-C [Earliest Transmissions to Slave]	697
4.4.2.7	LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]	699
4.4.2.8	LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]	700
4.4.2.9	LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update].....	701
4.4.2.10	LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy].....	703
4.4.3	MAS	704
4.4.3.1	LL/TIM/MAS/BV-01-C [Earliest Transmissions to Master]	704
4.4.3.2	LL/TIM/MAS/BV-02-C [Latest Transmissions to Master]	705
4.4.3.3	LL/TIM/MAS/BV-03-C [Initiate Sleep Clock Accuracy Update]	706
4.4.3.4	LL/TIM/MAS/BV-04-C [Response to Sleep Clock Accuracy Update].....	708
4.4.3.5	LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy].....	709
4.4.3.6	LL/TIM/MAS/BV-06-C [Change Accuracy after Update]	711
4.4.3.7	LL/TIM/MAS/BV-07-C [Change Accuracy before Update]	712
4.4.4	SCN	714
4.4.4.1	LL/TIM/SCN/BV-01-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner]	714
4.4.4.2	LL/TIM/SCN/BV-02-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY]	716
4.4.4.3	LL/TIM/SCN/BV-03-C [Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY]	718
4.5	ENC	720
4.5.1	ADV	720
4.5.1.1	LL/ENC/ADV/BI-01-C [Scan Request Invalid Address]	720
4.5.1.2	LL/ENC/ADV/BI-02-C [Master Packets Invalid Address]	723
4.5.2	SCN	725
4.5.2.1	LL/ENC/SCN/BI-01-C [Passive Scanning Invalid Address]	725
4.5.2.2	LL/ENC/SCN/BI-02-C [Active Scanning Invalid Address]	727
4.5.3	INI	729
4.5.3.1	LL/ENC/INI/BI-01-C [Slave Packets Invalid Address]	729
4.5.4	SLA	731
4.5.4.1	LL/ENC/SLA/BI-01-C [Packets to another Slave]	731
4.5.5	MAS	733
4.5.5.1	LL/ENC/MAS/BI-01-C [Packets to another Master]	733
4.6	FRH.....	735
4.6.1	Common PDU Contents	735
4.6.2	ADV	735
4.6.2.1	LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]	735
4.6.3	SLA	737
4.6.3.1	LL/FRH/SLA/BV-01-C [Accepting Channel Map Update]	737
4.6.3.2	LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2]	739
4.6.4	MAS	740
4.6.4.1	LL/FRH/MAS/BV-01-C [Requesting Channel Map Update].....	740



4.6.4.2	LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2]	742
4.6.4.3	LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels]	744
4.7	PAC.....	745
4.7.1	Common PDU Contents	745
4.7.2	SLA	745
4.7.2.1	LL/PAC/SLA/BI-01-C [Control PDUs with Invalid Length from Master].....	745
4.7.2.2	LL/PAC/SLA/BV-01-C [Unknown Packet from Master].....	746
4.7.3	MAS.....	747
4.7.3.1	LL/PAC/MAS/BI-01-C [Control PDUs with Invalid Length from Slave].....	747
4.7.3.2	LL/PAC/MAS/BV-01-C [Unknown Packet from Slave].....	748
4.8	SEC.....	749
4.8.1	Common PDU Contents	750
4.8.2	ADV	750
4.8.2.1	LL/SEC/ADV/BV-01-C [Advertising With Static Address]	750
4.8.2.2	LL/SEC/ADV/BV-02-C [Privacy - Non-connectable Undirected Advertising, non- resolvable private address].....	752
4.8.2.3	LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address].....	754
4.8.2.4	LL/SEC/ADV/BV-04-C [Network Privacy - Scannable Advertising, non-resolvable private address].....	756
4.8.2.5	LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address].....	758
4.8.2.6	LL/SEC/ADV/BV-06-C [Network Privacy - Undirected Connectable Advertising no Local IRK, no peer IRK]	760
4.8.2.7	LL/SEC/ADV/BV-07-C [Network Privacy - Undirected Connectable Advertising with Local IRK, no peer IRK]	762
4.8.2.8	LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK].....	764
4.8.2.9	LL/SEC/ADV/BV-09-C [Network Privacy - Undirected Connectable Advertising without Local IRK and with peer IRK].....	766
4.8.2.10	LL/SEC/ADV/BV-10-C [Network Privacy - Undirected Connectable Advertising using Resolving List and Peer Device Identity not in the List].....	768
4.8.2.11	LL/SEC/ADV/BV-11-C [Network Privacy - Directed Connectable Advertising using local and remote IRK]	770
4.8.2.12	LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK]	772
4.8.2.13	LL/SEC/ADV/BV-13-C [Network Privacy - Directed Connectable Advertising without local IRK but with remote IRK]	774
4.8.2.14	LL/SEC/ADV/BV-14-C [Network Privacy - Directed Connectable Advertising using Resolving List and Peer Device Identity not in the List].....	776
4.8.2.15	LL/SEC/ADV/BV-15-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore Identity Address].....	778
4.8.2.16	LL/SEC/ADV/BV-16-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Ignore Identity Address]	780
4.8.2.17	LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address].....	782
4.8.2.18	LL/SEC/ADV/BV-18-C [Device Privacy - Scannable Advertising, resolvable private address, Accept Identity Address]	784
4.8.2.19	LL/SEC/ADV/BV-19-C [Device Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Accept Identity Address].....	786
4.8.2.20	LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address].....	788



4.8.2.21	LL/SEC/ADV/BV-21-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA].....	790
4.8.3	SCN	792
4.8.3.1	LL/SEC/SCN/BV-01-C [Random Address Scanning]	792
4.8.4	INI	794
4.8.5	SLA	794
4.8.5.1	LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup].....	794
4.8.5.2	LL/SEC/SLA/BV-02-C [Slave Pause Encryption].....	797
4.8.5.3	LL/SEC/SLA/BV-03-C [Slave Pause Encryption Sending Data]	799
4.8.5.4	LL/SEC/SLA/BV-04-C [Slave Sending Reject_Ind].....	801
4.8.5.5	LL/SEC/SLA/BV-05-C [Slave Receiving Encrypted Data]	802
4.8.5.6	LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled]	804
4.8.5.7	LL/SEC/SLA/BV-07-C [Responding to LL_PING_REQ]	805
4.8.5.8	LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ]	806
4.8.5.9	LL/SEC/SLA/BV-09-C [Modified Authentication Payload Timeout].....	808
4.8.5.10	LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]	809
4.8.5.11	LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND].....	811
4.8.5.12	LL/SEC/SLA/BI-01-C [Slave Encryption Setup: Missing Response].....	812
4.8.5.13	LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC]	814
4.8.5.14	LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header]	815
4.8.5.15	LL/SEC/SLA/BI-05-C [Slave Receiving unexpected PDU during encryption start]	816
4.8.6	MAS	818
4.8.6.1	LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup].....	818
4.8.6.2	LL/SEC/MAS/BV-02-C [Master Pause Encryption].....	820
4.8.6.3	LL/SEC/MAS/BV-03-C [Master Receiving Reject_Ind]	822
4.8.6.4	LL/SEC/MAS/BV-04-C [Master Encryption: Sending Data before LL_ENC_RSP]	823
4.8.6.5	LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]	825
4.8.6.6	LL/SEC/MAS/BV-06-C [Initiate LE Ping procedure when encryption is enabled]	827
4.8.6.7	LL/SEC/MAS/BV-07-C [Responding to LL_PING_REQ]	828
4.8.6.8	LL/SEC/MAS/BV-08-C [No response to LL_PING_REQ]	829
4.8.6.9	LL/SEC/MAS/BV-09-C [Modified Authentication Payload Timeout].....	831
4.8.6.10	LL/SEC/MAS/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]	832
4.8.6.11	LL/SEC/MAS/BV-11-C [Master Receiving LL_REJECT_EXT_IND]	834
4.8.6.12	LL/SEC/MAS/BV-12-C [Master Start Encryption: Overlapping Procedure].....	835
4.8.6.13	LL/SEC/MAS/BV-13-C [Master Start Encryption: Overlapping Procedure with LL_SLAVE_FEATURE_REQ].....	837
4.8.6.14	LL/SEC/MAS/BV-14-C [Master Receiving unexpected PDU during encryption start]	838
4.8.6.15	LL/SEC/MAS/BI-01-C [Master Encryption Setup: Missing Response].....	840
4.8.6.16	LL/SEC/MAS/BI-03-C [Master Encryption Setup: Missing Request].....	841
4.8.6.17	LL/SEC/MAS/BI-04-C [Master Encryption Setup: Missing Acknowledgement].....	842
4.8.6.18	LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC]	843
4.8.6.19	LL/SEC/MAS/BI-06-C [Master MIC Failure: Corrupt Header]	845
4.8.6.20	LL/SEC/MAS/BI-07-C [Master Pause Encryption Sending Data]	846
4.8.6.21	LL/SEC/MAS/BI-08-C [Master Encryption: Sending Data and Not Response]	848
4.8.6.22	LL/SEC/MAS/BI-09-C [Master Encryption: Sending Data and Not Request]	849
4.9	Data Flow.....	851
4.9.1	Both Connected Roles	851



4.9.1.1	[Transmit Fragmented L2CAP Header]	851
4.9.1.2	[Receive Fragmented L2CAP Header]	853
5	Test Case Mapping.....	856



1 Scope

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

The objective of this test suite is to provide a basis for 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. For the purpose of this Bluetooth document, the definitions and abbreviations found in [1], [2], and [3] apply.

- [1] Specification of the Bluetooth System, Versions 4.0 or later
- [2] Bluetooth Test Strategy and Terminology Overview
- [3] Specification of the Bluetooth System, Volume 6 Part B (Link Layer Protocol Specification), Version 4.0
- [4] ICS Proforma for Bluetooth Low Energy Link Layer Protocol Specification
- [5] ITU-T X.290 series, OSI CONFORMANCE TESTING METHODOLOGY AND FRAMEWORK PROTOCOL RECOMMENDATIONS FOR ITU-T APPLICATIONS, ITU Recommendation X.290 series (equivalent to ISO 9646)
- [6] Federal Information Processing Standards Publication 197, ADVANCED ENCRYPTION STANDARD (AES), FIPS, 2001
- [7] Bluetooth Test Suite for RF-PHY
- [8] Specification of the Bluetooth System, Volume 6 Part B (Link Layer Protocol Specification), Version 4.2 or later
- [9] Bluetooth Core IXIT proforma
- [10] Specification of the Bluetooth System, Volume 6 Part B (Link Layer Protocol Specification), Version 5.0 or later
- [11] Specification of the Bluetooth System, Volume 2, Part E Host Controller Interface (HCI)
- [12] Specification of the Bluetooth System, Volume 3, Part A, Logical Link Control and Adaptation Protocol (L2CAP) Specification
- [13] Specification of the Bluetooth System, Volume 6 Part B (Link Layer Protocol Specification), Version 5.1 or later



3 Test Suite Structure (TSS)

3.1 Test Strategy

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

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

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

The bullet points in [Figure 3.1](#) outline the test functionality grouped according to protocol procedures, orthogonal functions and testing objectives. Features and roles subject to testing appear in the groups as subgroups or functions.



- Device Discovery
 - Advertising
 - Non connectable Events
 - Connectable Events
 - Device Filtering
 - Advertising Data
 - Invalid Request Packets
 - Continuous Advertising
 - Combining Roles and Advertising
 - Scanning
 - Passive
 - Active
 - Device Filtering
 - Invalid Advertising Packets
 - Continuous Scanning
 - Combining Roles and Scanning
- Connection Handling
 - Advertising
 - Accepting Connections
 - Invalid Packets from Initiators
 - Initiating
 - Initiating
 - Invalid Packets from Advertisers and Slave
 - Addressing
 - Slave
 - Acknowledgement Scheme
 - Event Handling
 - Addressing
 - Invalid Packets from Master
 - Data Transfer
 - Accepting Parameter Update
 - Termination
 - Combining Slave Role
 - Initiate and Accept Parameter Request
 - Master
 - Acknowledgement Scheme
 - Addressing
 - Invalid Packets from Slave
 - Data Transfer
 - Parameter Update
 - Termination
 - Combining Master Role
 - Initiate and Accept Parameter Request
- Timing in Active and Low Power Modes
- Radio Frame Encoding
 - Incorrect Frames
- Frequency Hopping
- Packet Formats
- Security
 - Private Addressing
 - Encryption Mode Change

Figure 3.1: Test Suite Structure

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

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



3.2 Test Groups

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

3.2.1 Device Discovery

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

3.2.2 Connection Handling

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

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

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

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

3.2.3 Timing

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

3.2.4 Radio Frame Encoding

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

3.2.5 Frequency Hopping

Test the data channel selection algorithm.

3.2.6 Packet Formats

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

3.2.7 Security

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

3.3 Behavior Tests

The tests include both valid and invalid behavior as described by the protocol requirements. In valid behavior tests the IUT is triggered with valid message sequences and is expected to perform the protocol function. In invalid behavior tests the IUT is triggered with sequences containing invalid messages,



missing messages, messages of incorrect type or content and is expected to recover and resume the protocol function.

3.3.1 Valid Behavior (BV) Tests

These tests verify that the IUT reacts in conformity with the specification when receiving with valid message sequences.

3.3.2 Invalid Behavior (BI) Tests

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

3.4 Test Realization

An Upper Tester interface is required in the conformance testing configuration. A subset of the HCI [1] Vol 2 Part D is used as the TCI. In test realization, an Implementation under Test (“IUT”) is assumed to implement the HCI commands and events either explicitly or implicitly referenced in a supported procedure requirement in the protocol specification.



4 Test Cases (TC)

4.1 Introduction

4.1.1 Test Case Naming Conventions

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

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

Identifier Abbreviation	Spec Abbreviation Identifier <spec abbreviation>
LL	Link Layer
Identifier Abbreviation	Feature Identifier <feat>
CON	Connection Handling
DDI	Device Discovery
ENC	Radio Frame Encoding
FRH	Frequency Hopping
PAC	Packet Formats
SEC	Security
TIM	Timing
Identifier Abbreviation	Function Identifier <func>
ADV	Advertiser role
INI	Initiator role
MAS	Master role
SCN	Scanner role
SLA	Slave role

Table 4.1: TC Feature naming convention for LL

4.1.2 Conformance

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

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

Such tests may verify:

- That claimed capabilities may be used in any order and any number of repetitions that is not excluded by the Specification, OR



- That capabilities enabled by the implementations are sustained over durations expected by the use case, OR
- That the implementation gracefully handles any quantity of data expected by the use case, OR
- That in cases where more than one valid interpretation of the Specification exist, the implementation complies with at least one interpretation and gracefully handles other interpretations OR
- That the implementation is immune to attempted security exploits.

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

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

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

4.1.3 Common Packet Contents

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

- The Least Significant Bit (LSB) corresponds to b_0 ;
- The LSB is the first bit sent over the air;

In illustrations, the LSB is shown on the left side.

4.1.3.1 Access Address Used

When the IUT is in the advertising state or slave role, the default value for the data channel access address is 0x456789AB. With the same convention the advertising channel access address (see below) is expressed with a hexadecimal number as 0x8E89BED6.

Where an invalid access address is used for a frame, it is corrupted by inverting a single bit and the test procedure steps will by repetition reflect the probability of the error being reverted. The corruption is done alternating from the last bit before the PDU and from the second bit after the preamble, such that the addresses used are invalid data for the tests but still according to the protocol transfer syntax.

4.1.4 Common Message Contents

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

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

4.1.4.1 Checksums Used

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



4.1.4.2 Device Addresses Used

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

When it is required to modify the Lower Tester address either in the company_assigned field, the company_id field, it has to be done by exchanging bytes 1 and 3 of company_assigned field or company_id field. If it is required to modify the Lower Tester address in the company_assigned and company_id fields, it has to be done by exchanging bytes 1 of company_assigned and company_id fields.

4.1.5 Common Initial and Final Conditions

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

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

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

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

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

If a test does not mention unmasking events, the IUT shall unmask all events mentioned in that test before the test procedure is executed.

If a test covers optional features and does not mention the feature exchange procedure, the IUT shall perform the feature exchange procedure before the test procedure is executed.

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



State: Standby

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

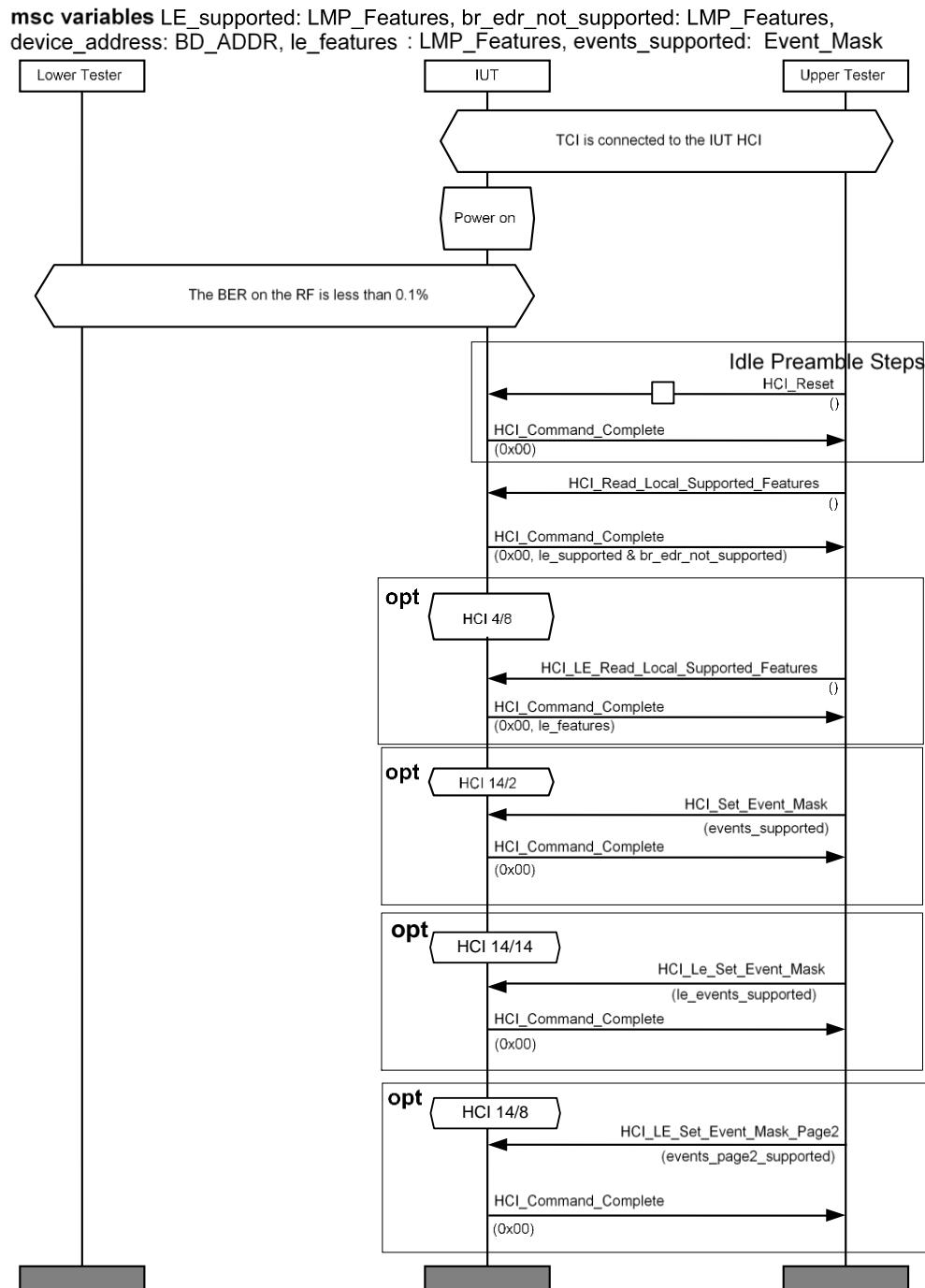


Figure 4.1: Standby Preamble Steps

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



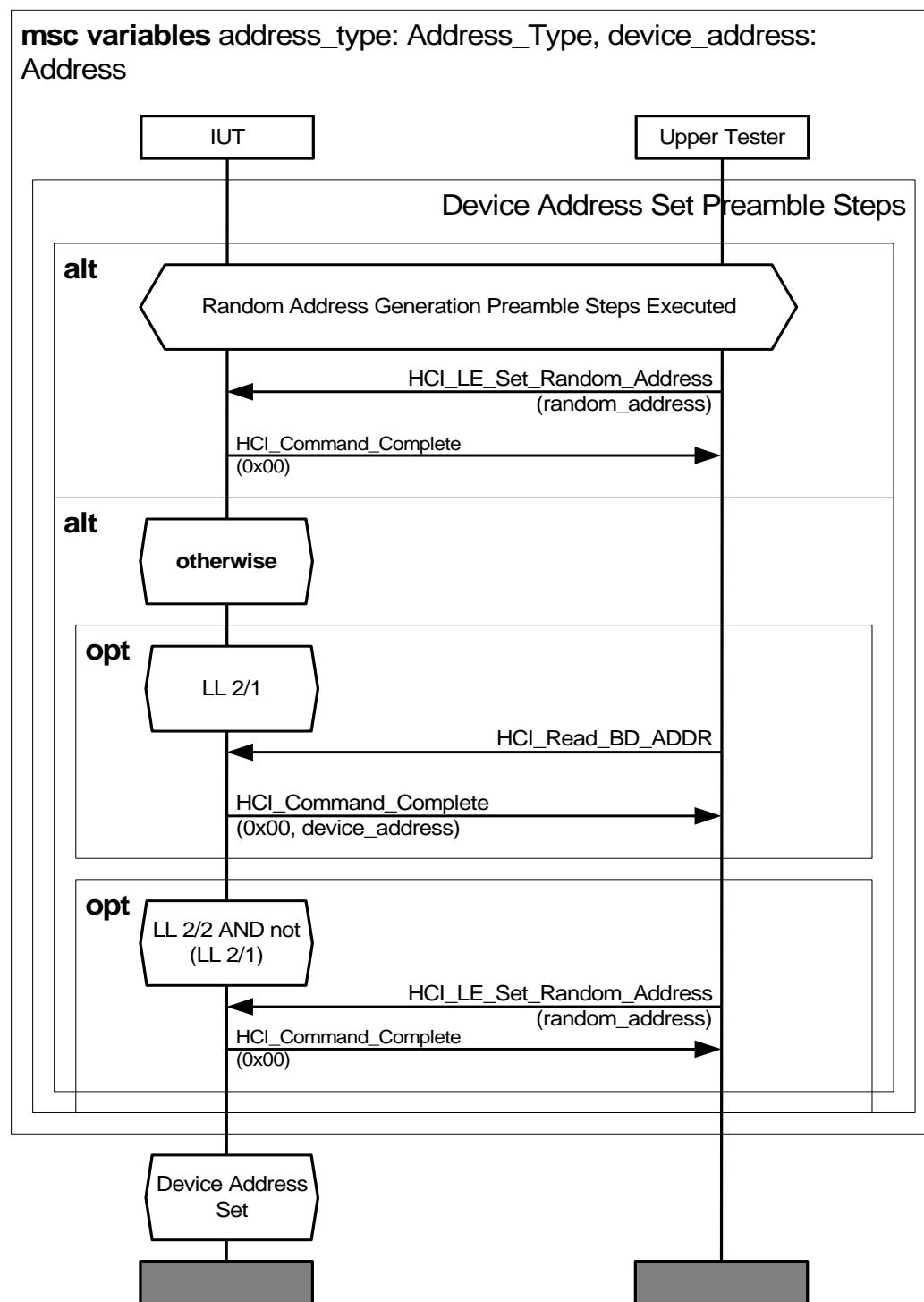
State: Device Address Set

Figure 4.2: Device Address Set Preamble Steps

The preamble steps above read or set an address supported by the IUT.



State: Specific White Listed

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

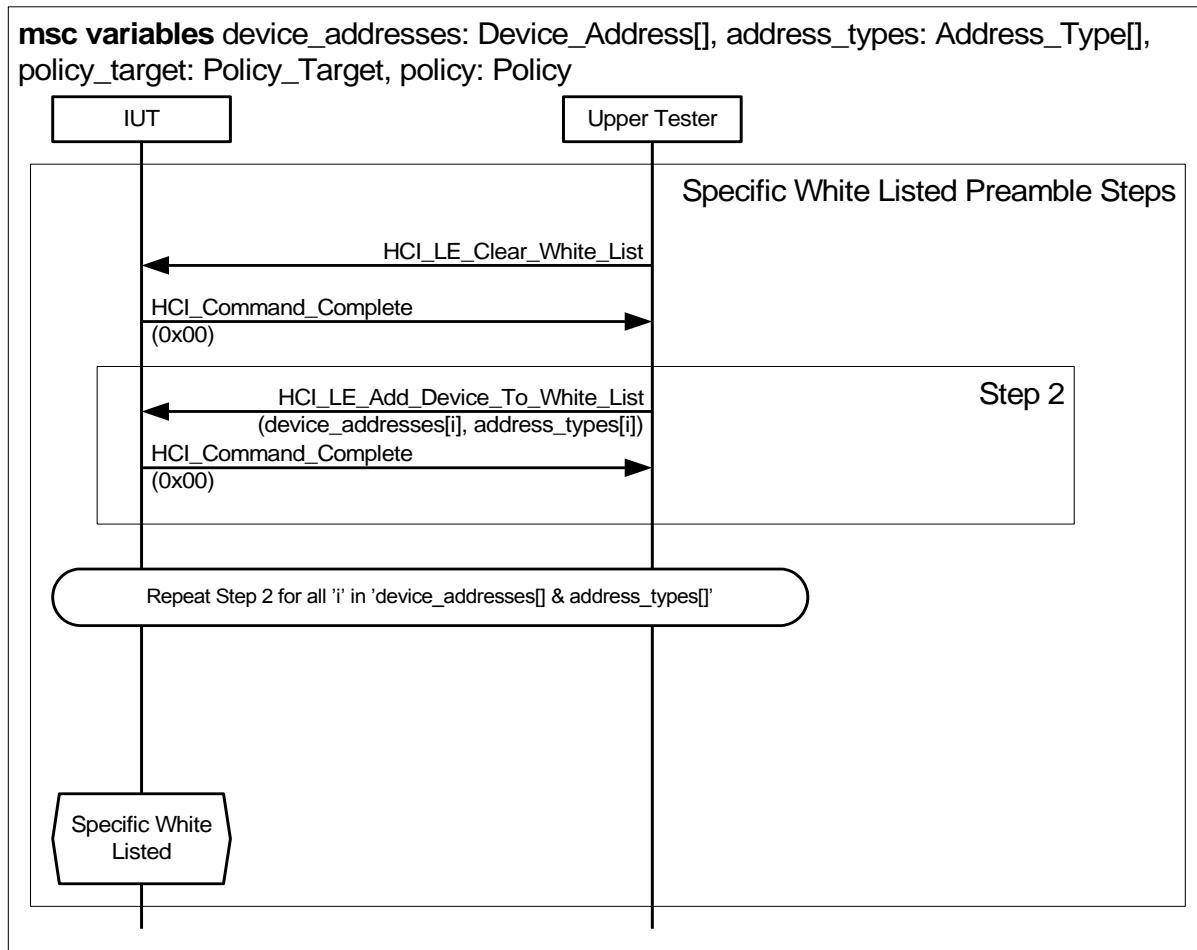


Figure 4.3: Specific White Listed Preamble Steps

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



State: Buffer Size Read

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

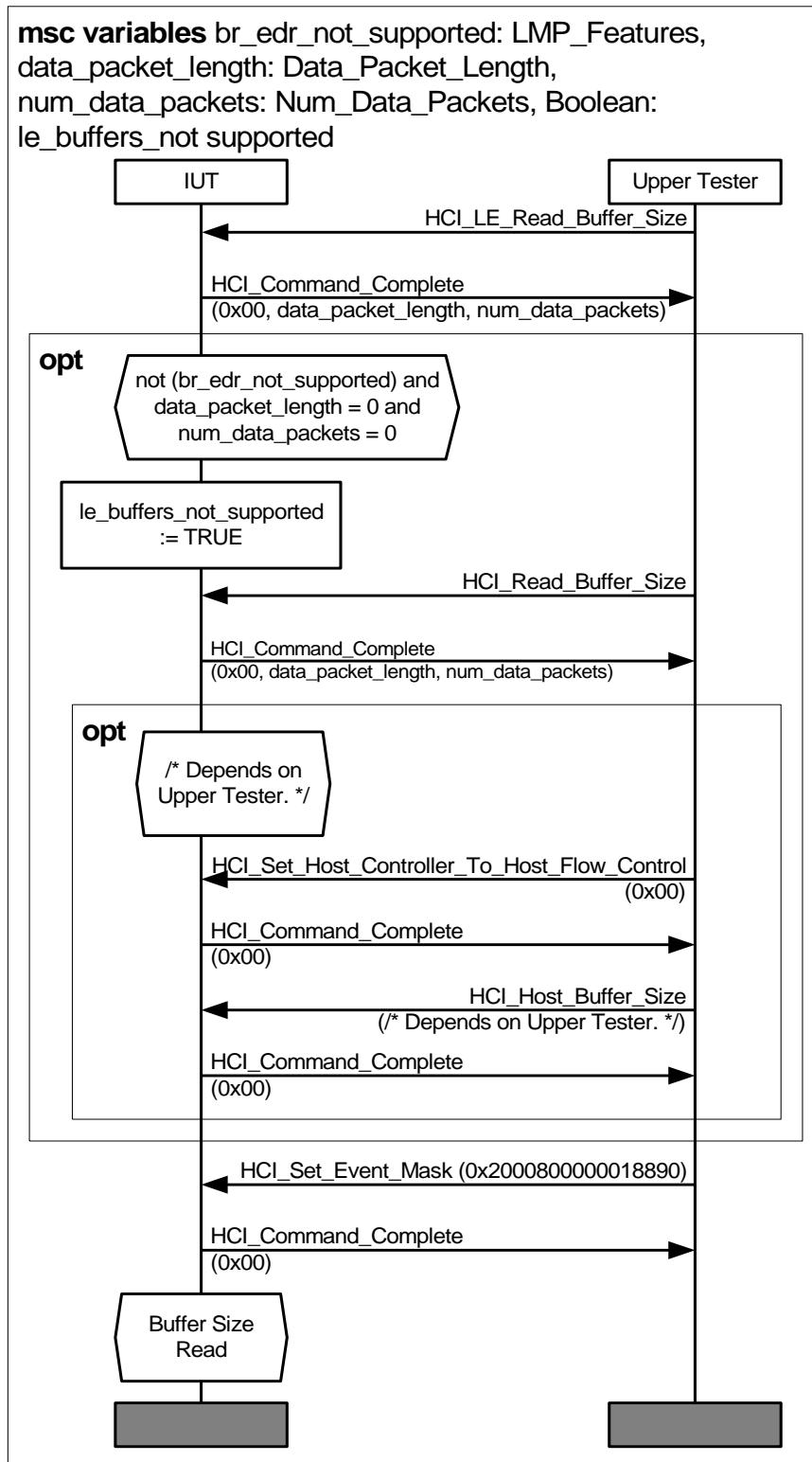


Figure 4.4: Buffer Size Read Preamble Steps



The outcome of the preamble steps is the information about data primitives and parameters supported by the IUT as well as the configuration of the data primitive parameters supported by the Upper Tester to the IUT.

4.1.5.1 ADV

The common initial states of the advertising state.

State: Advertising Parameters Set

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

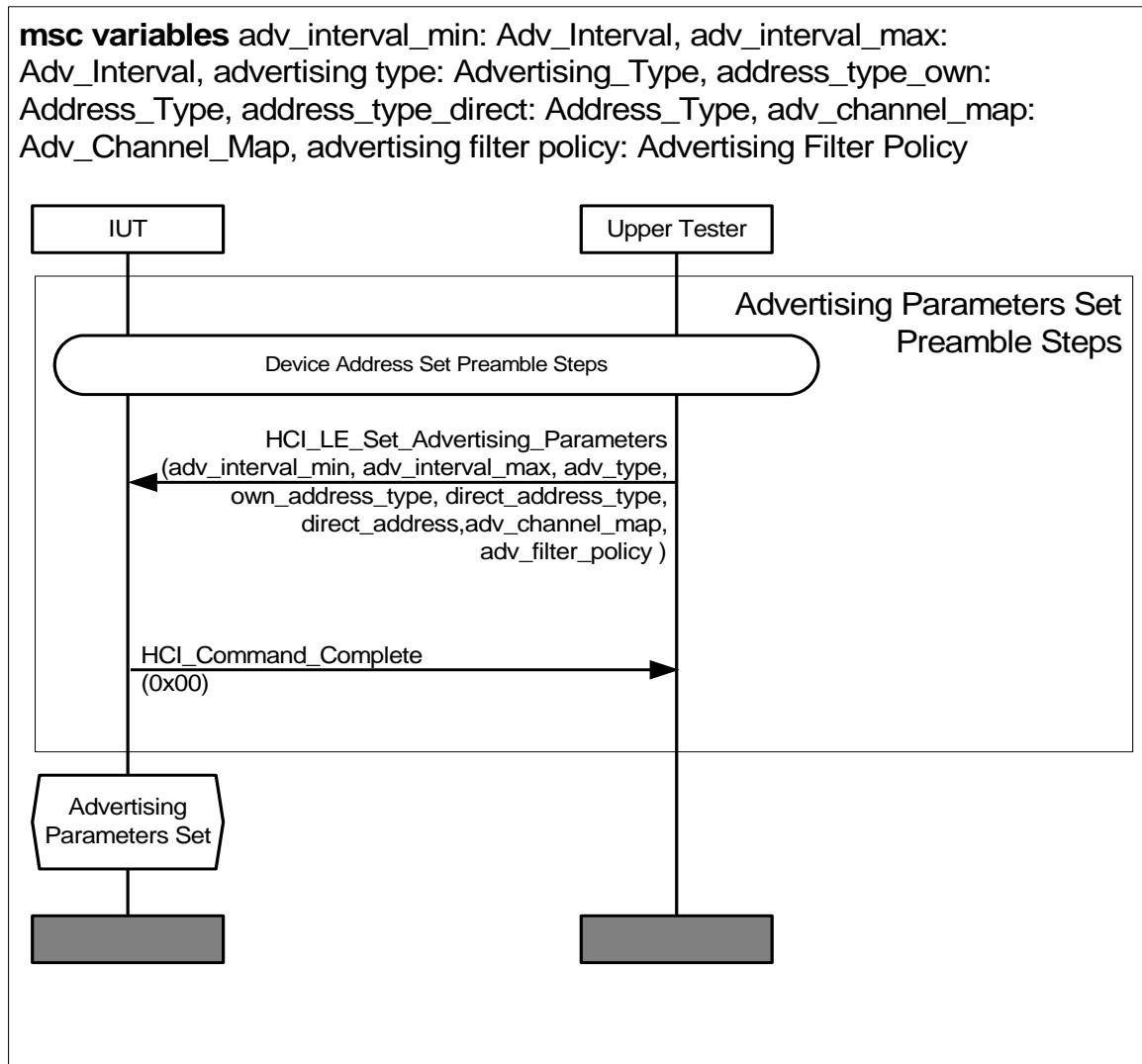


Figure 4.5: Advertising Parameters Set Preamble Steps



State: Non-Connectable Advertising

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

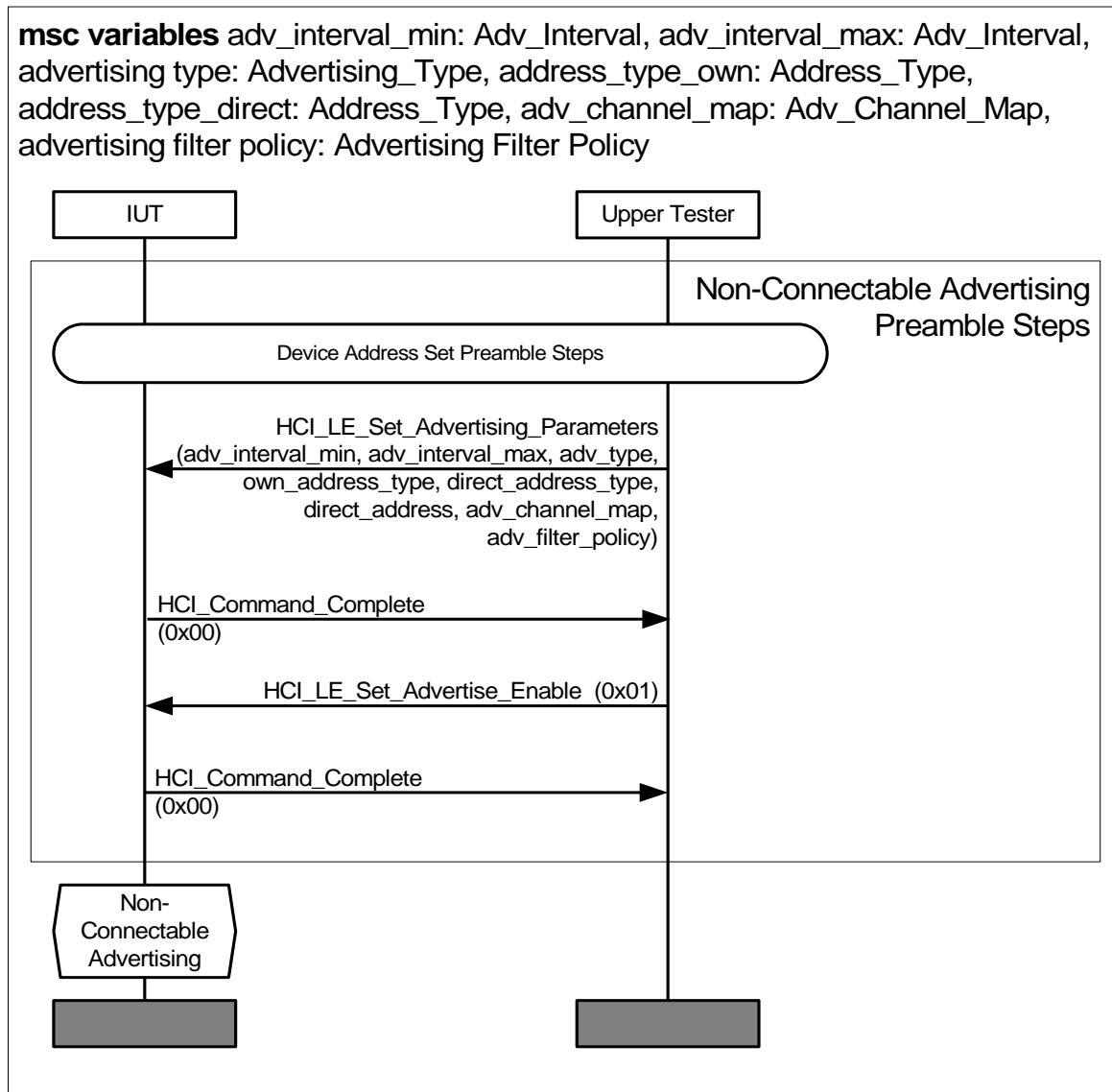


Figure 4.6: Non-Connectable Advertising Preamble Steps



State: Undirected Advertising

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

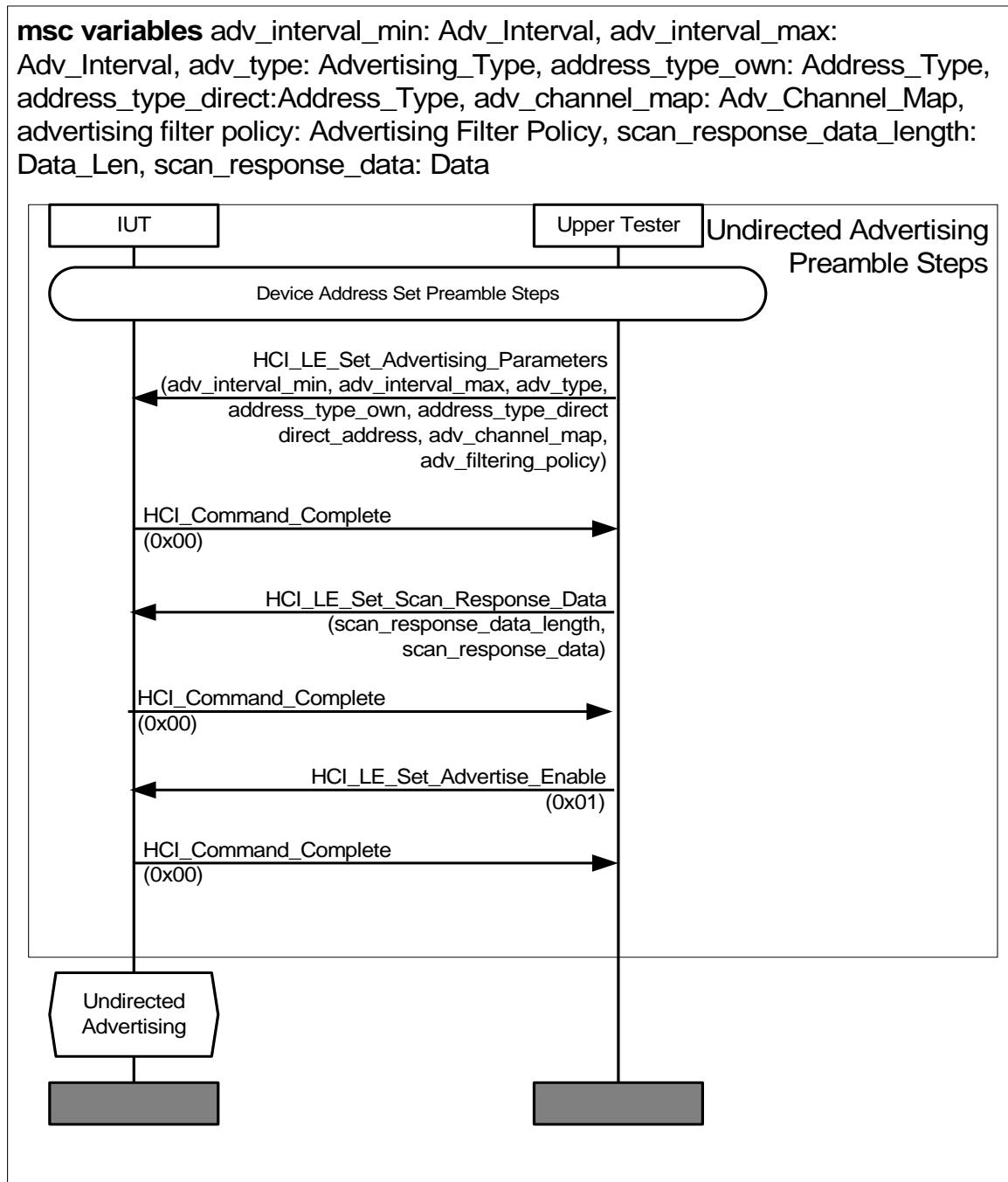


Figure 4.7: Undirected Advertising Preamble Steps



State: Discoverable Advertising

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

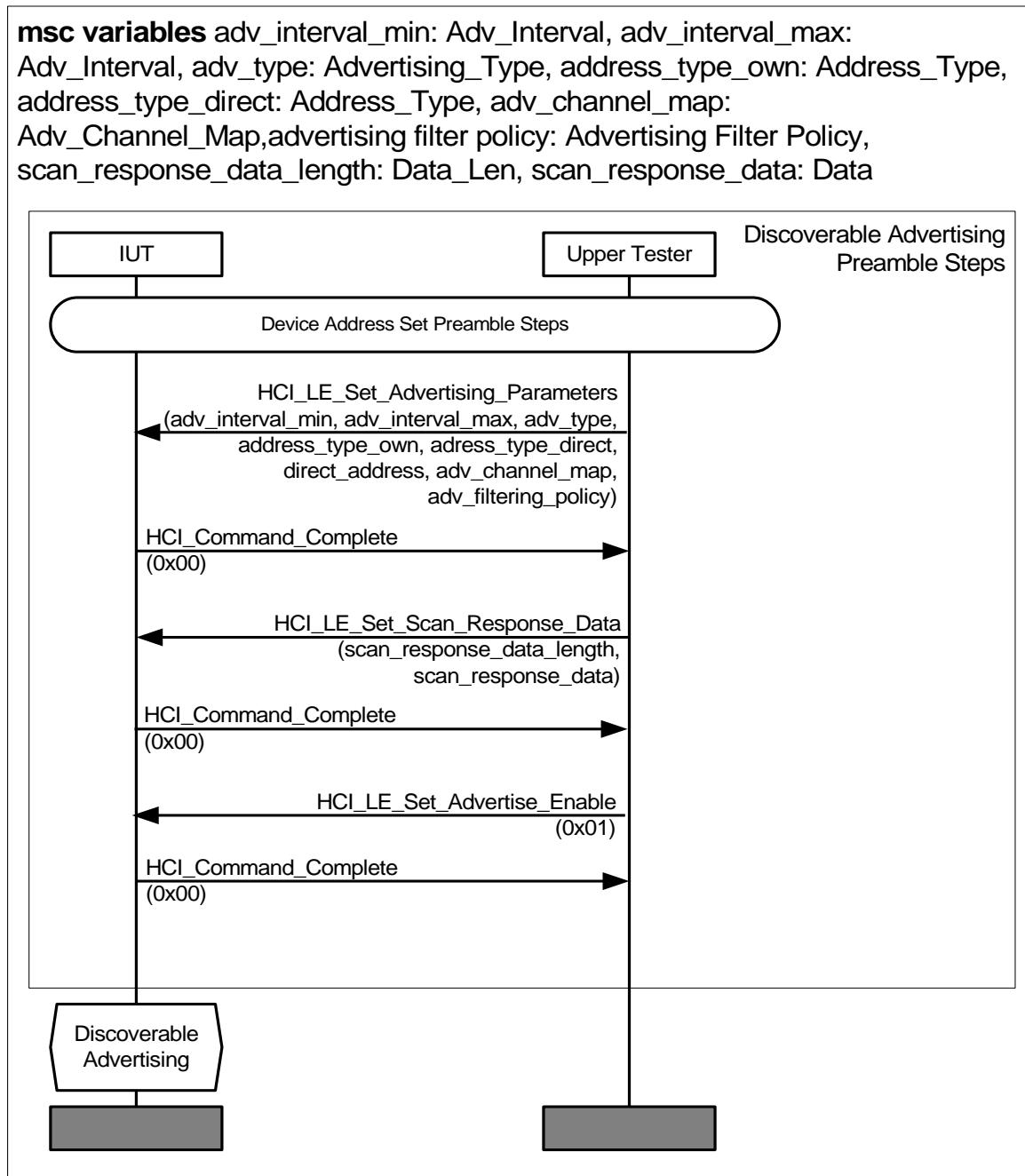


Figure 4.8: Discoverable Advertising Preamble Steps



State: Directed Advertising

The HCI commands to set the IUT to advertise using directed advertising with parameters events are displayed in the [Figure 4.9](#).

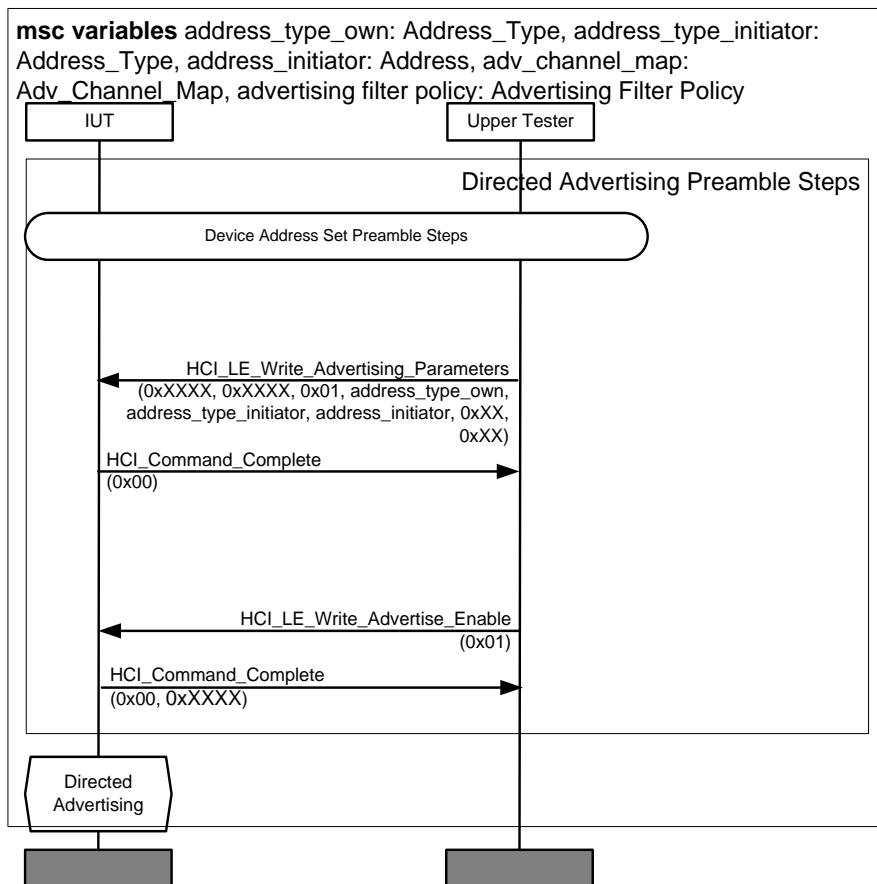


Figure 4.9: Directed Advertising Preamble Steps



4.1.5.2 SCN

The common initial states of the scanning state.

State: Passive Scanning

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

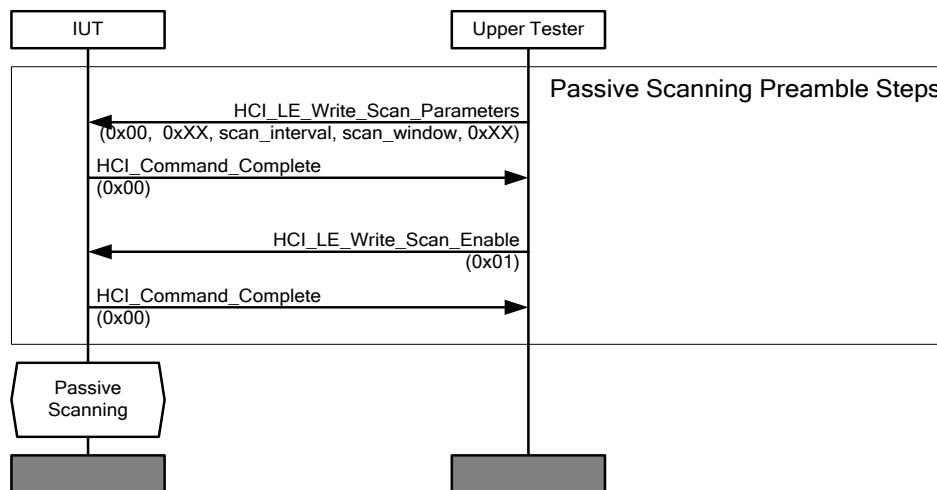


Figure 4.10: Passive Scanning Preamble Steps

State: Active Scanning

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

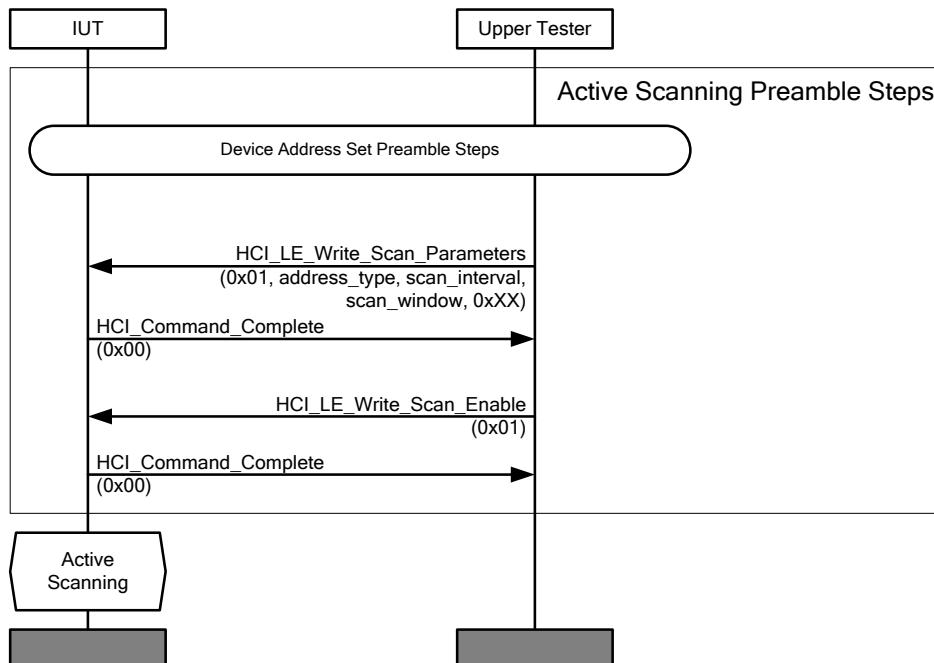


Figure 4.11: Active Scanning Preamble Steps



State: Low Duty Cycle Directed Advertising

The HCI commands to set the IUT to advertising using Low Duty Cycle Directed Advertising with parameters are displayed in [Figure 4.12](#).

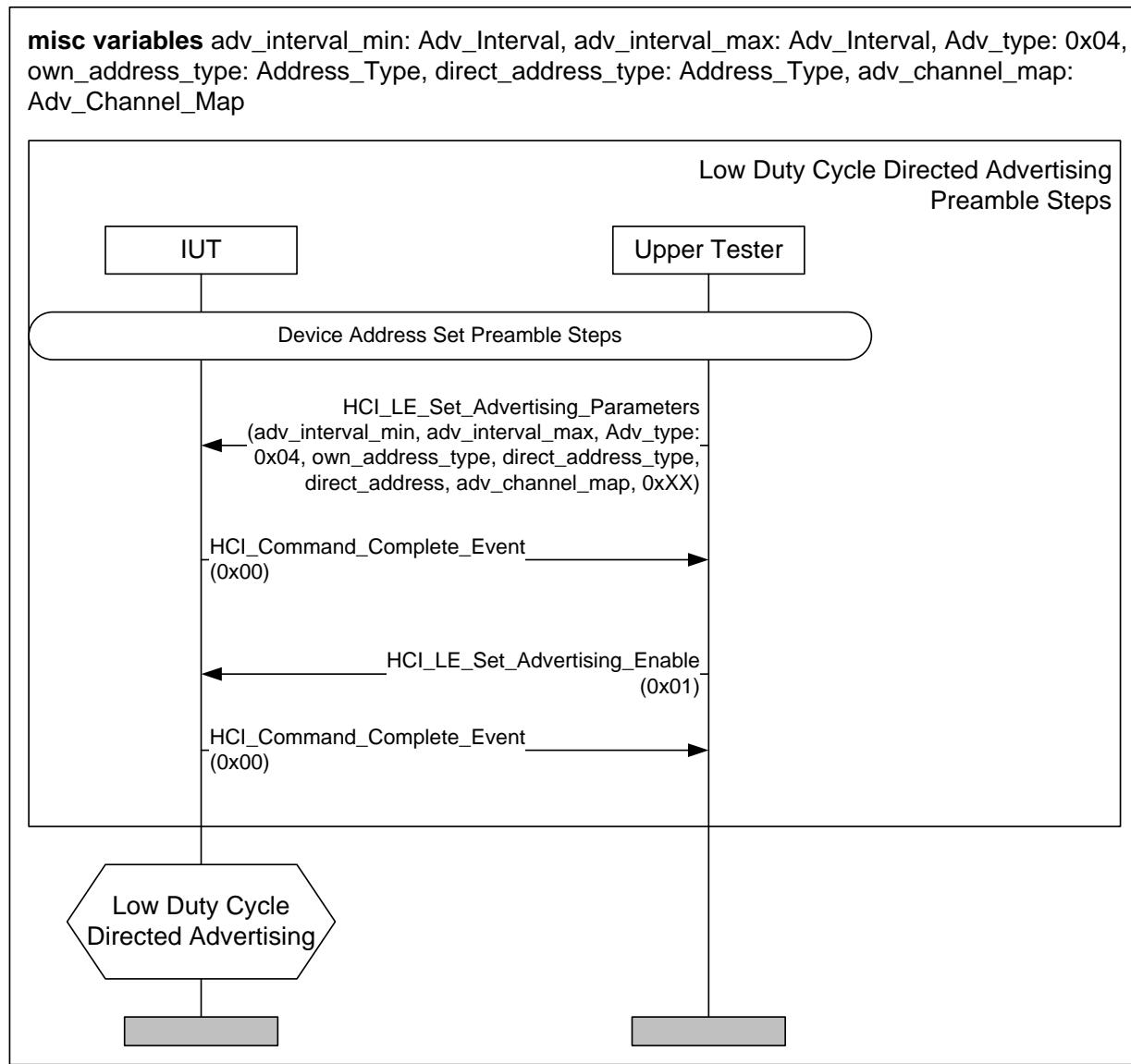


Figure 4.12: Low Duty Cycle Direct Advertising Preamble Steps

4.1.5.3 INI

The common initial states of the initiating state.



State: Initiating

The HCI commands to set the IUT to initiate a connection with parameters are displayed in [Figure 4.13](#).

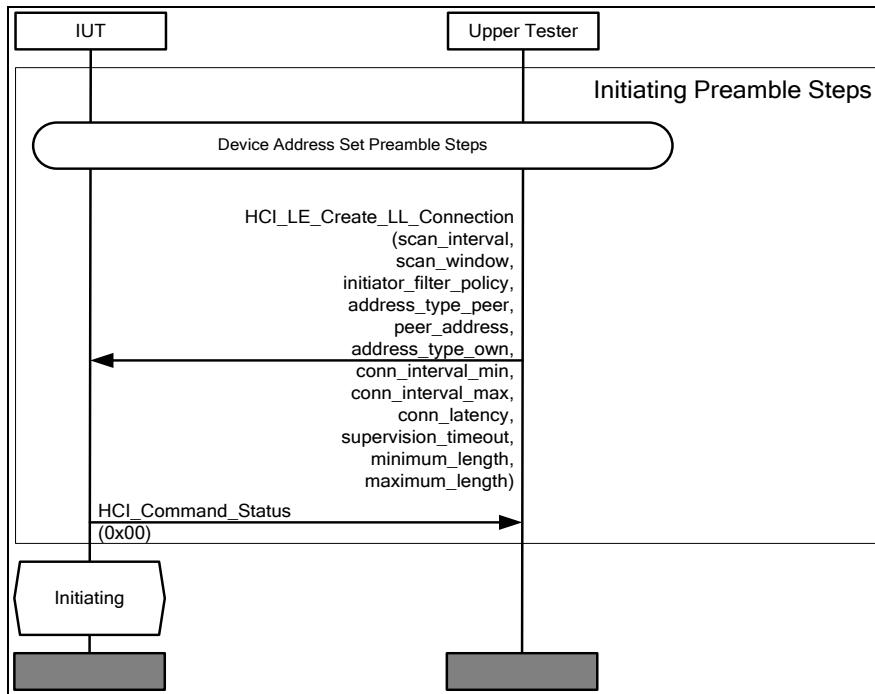


Figure 4.13: Initiating Preamble Steps

Note: peer_address is an address used by the Lower Tester.

State: Connection Setup White Listed

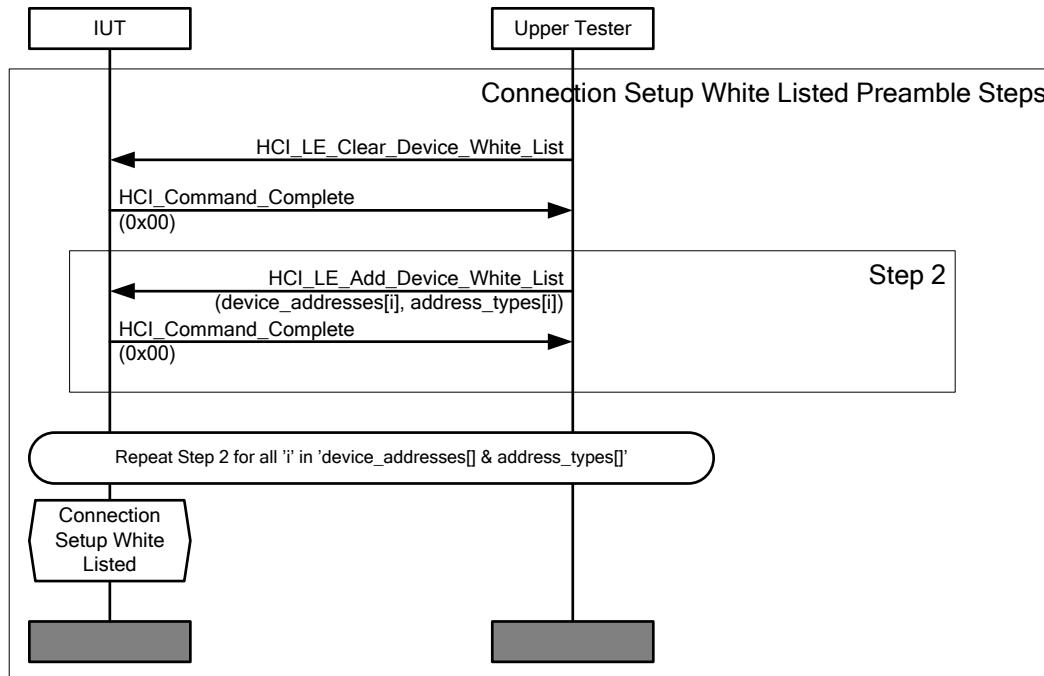


Figure 4.14: Connection Setup White Listed Preamble Steps

The initiator device filtering uses only the white listing mechanism.



4.1.5.4 SLA

The common initial states of the slave role.

State: Connected Slave

The HCI commands to set the IUT to accept a connection and start maintaining it in the slave role with parameters are displayed in [Figure 4.15](#).

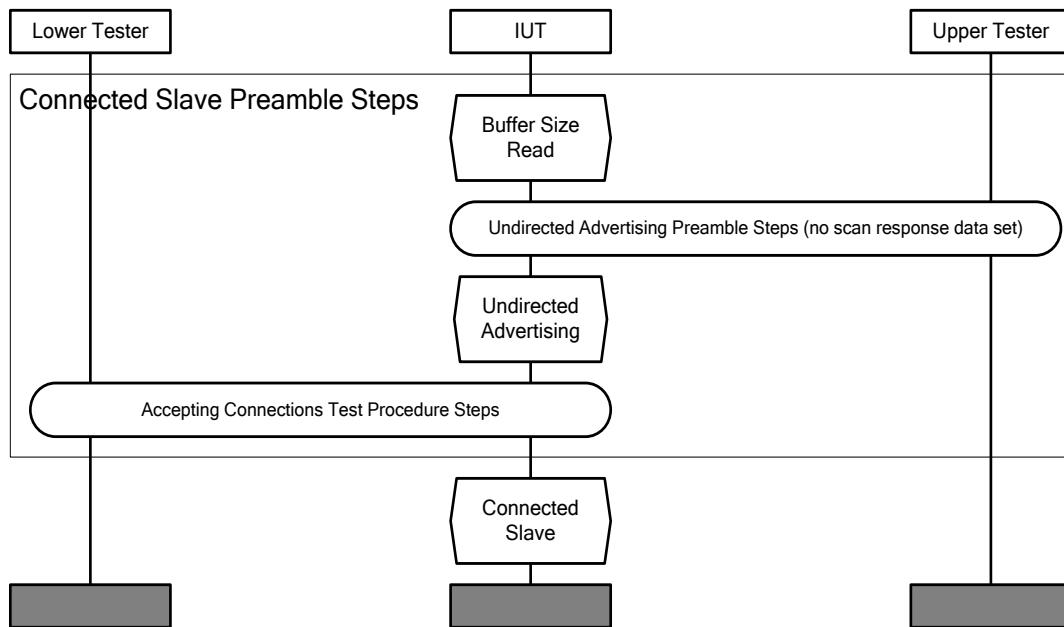


Figure 4.15: Connected Slave Preamble Steps

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



State: Slave Connection Terminated

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

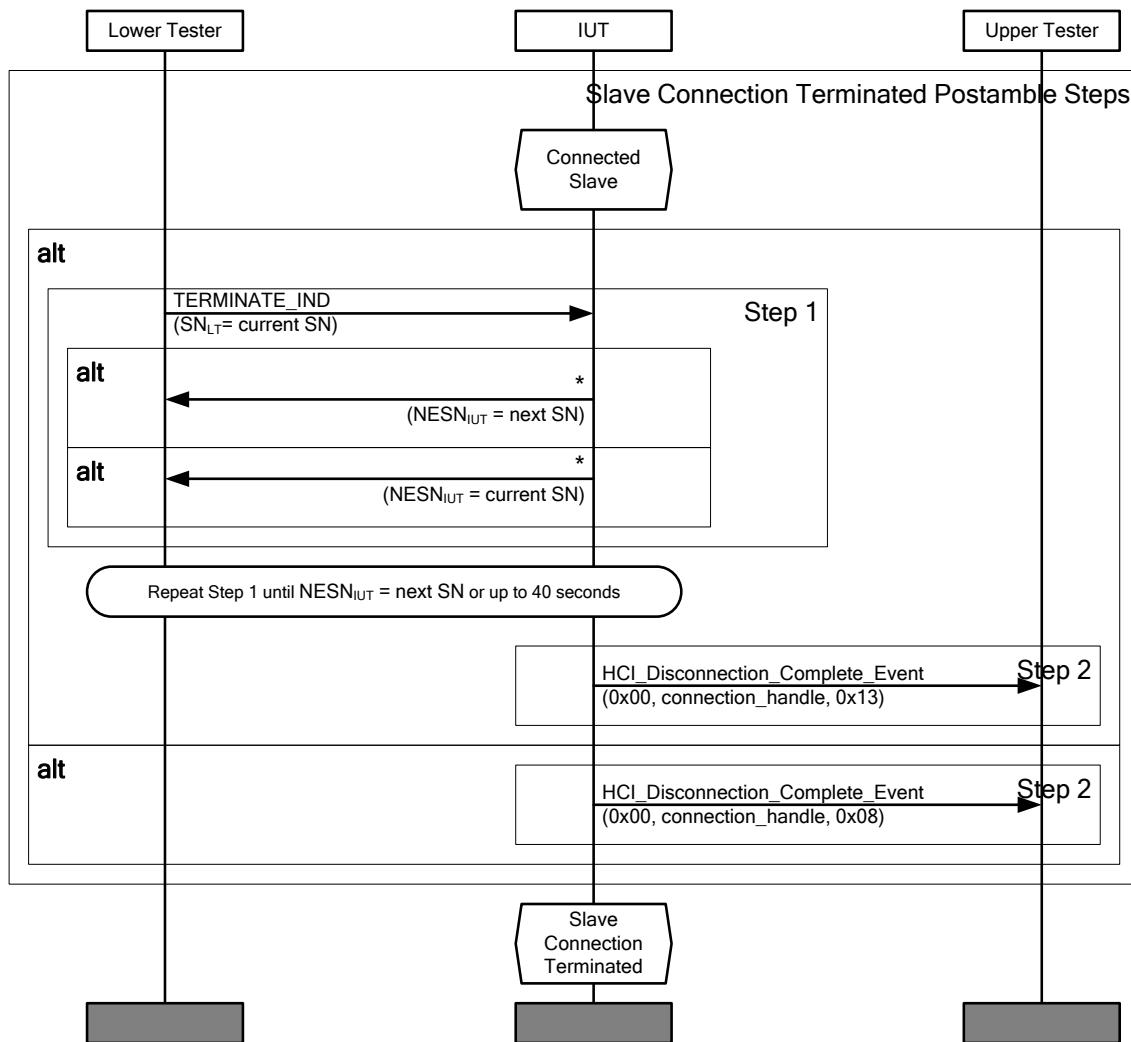


Figure 4.16: Slave Connection Terminated Postamble Steps

The asterisk denoting the packet type in Step 1 above means that only the acknowledgement scheme is followed in the steps and the packet types are not observed.



4.1.5.5 MAS

The common initial states of the master role.

State: Connected Master

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

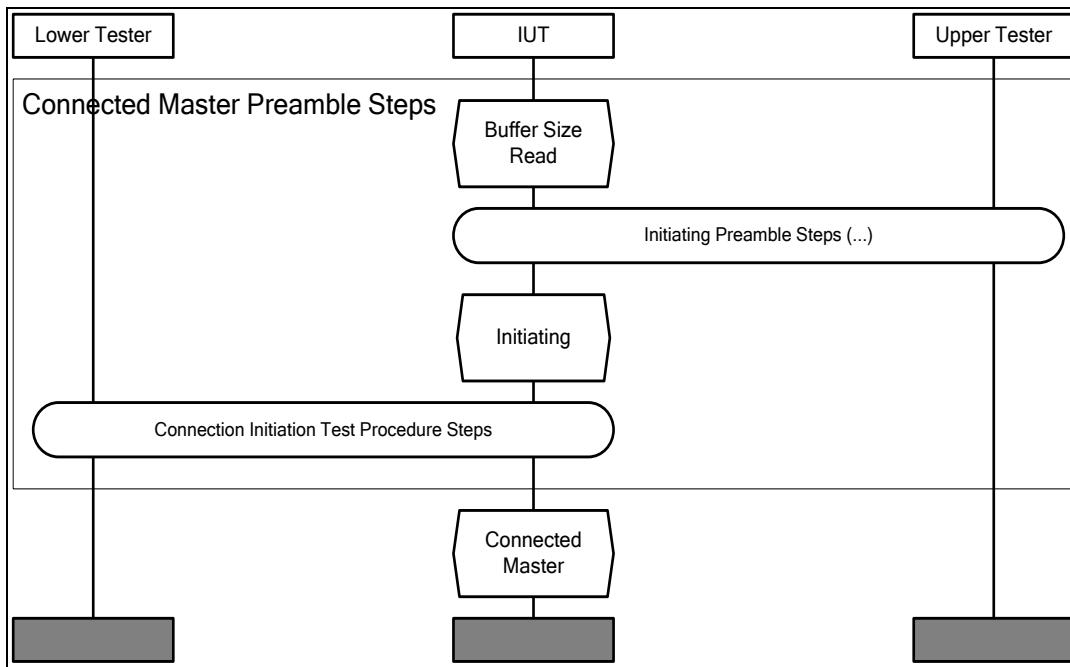


Figure 4.17: Connected Master Preamble Steps

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



State: Master Connection Terminated

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

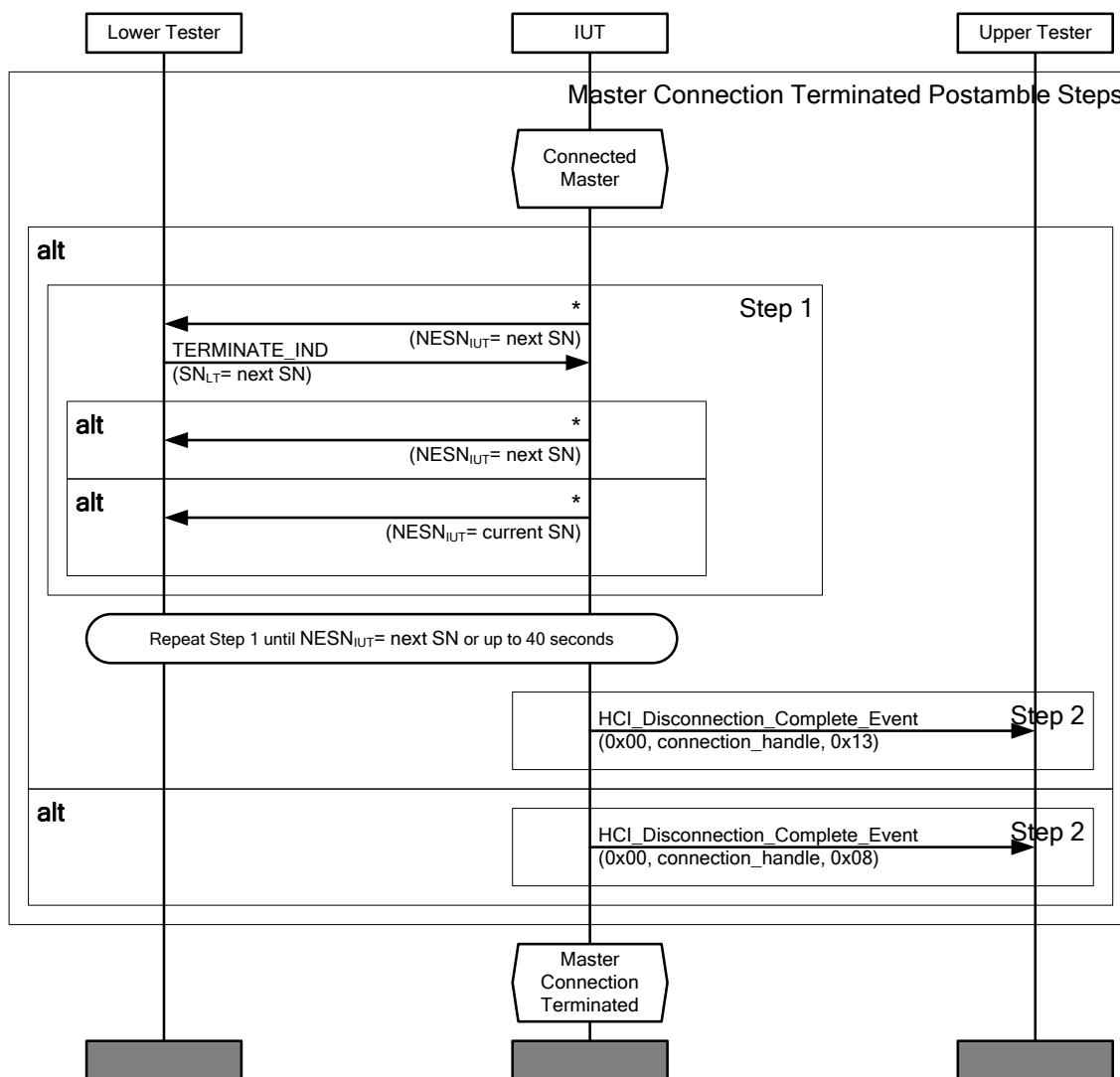


Figure 4.18: Master Connection Terminated Postamble Steps

The asterisk denoting the packet type in Step 1 above means that only the acknowledgement scheme is followed in the steps and the packet types are not observed.



4.1.5.6 SEC

State: Encrypted Address Calculated

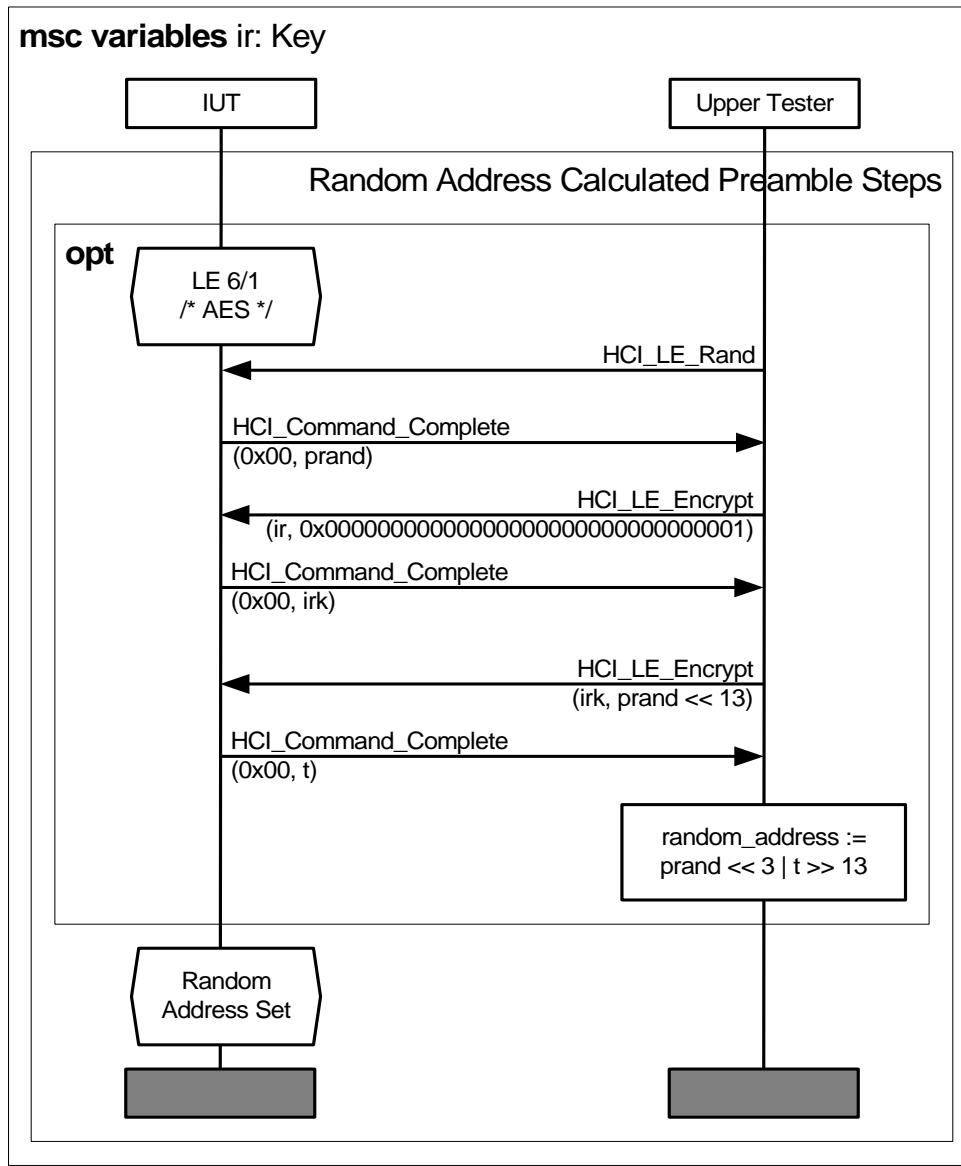


Figure 4.19: Random Address Calculated Preamble Steps

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



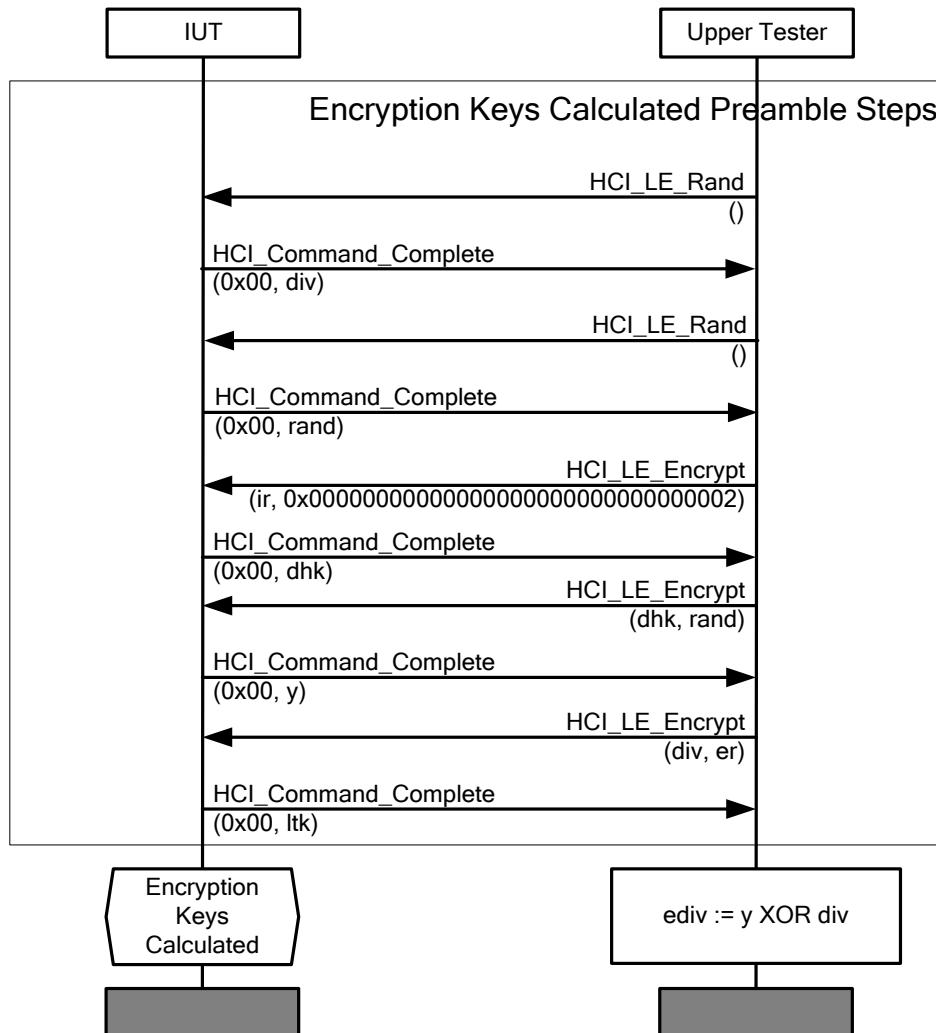
State: Encryption Keys Calculated

Figure 4.20: Encrypted Address Calculated Preamble Steps

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

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

Encryption keys are input to a Controller from [1] (part H, section 2.4.2). The keys to generate random addresses are obtained from the identity root "IR". The IR is referred to as 'ir' and has the default value 0x112233445566778899AABBCCDDEEFF00. The encryption root, ER, is referred to as 'er' and has the same default value. The key to encrypt LL data is referred to as 'ltk'.



4.1.5.7 Common Test Procedure Steps

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

4.1.5.7.1 Recovery Actions in Test Steps

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

- Step for expecting a packet, but receiving none:

Repeat the step expecting the packet a number of times to reflect the confidence in the verdict to assign. The test procedure may fail to execute after the repetition count or time interval, or a Fail verdict may be assigned with a total number of missing packets.

- Step for expecting a packet, but receiving a different packet with a correct CRC:

If the packet is from the IUT and matches the contents expected (advertising channels) or the timing and synchronization word (data channels), assume the IUT has sent the packet and assign a Pass verdict.

If there are no alternatives for the test procedure step and the testing limitations do not describe otherwise, assign a Fail verdict.

- Step for expecting a packet and receiving one with an incorrect CRC:

If the packet is from the IUT and matches the timing, repeat the step expecting the packet a number of times to reflect the confidence in the verdict to assign (Assuming that a packet from the IUT is corrupted). Stop the test procedure execution after the repetition count.

- Step for expecting no packet, but receiving one with a correct CRC:

If the packet received is from the IUT fail the test procedure step (Assuming the IUT has sent an inopportune packet).

- Step for expecting no packet, but receiving one with an incorrect CRC:

If the packet is from the IUT and matches the timing, fail the test procedure step (Assuming the IUT has sent an inopportune packet).

4.1.5.7.2 Test Step Repetition Count

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

The confidence level is proportional to the repetition count of the test.



4.1.5.7.3 Optional Test Steps

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

- Connection Update and Connection Parameters Request Procedures Optional Test Steps

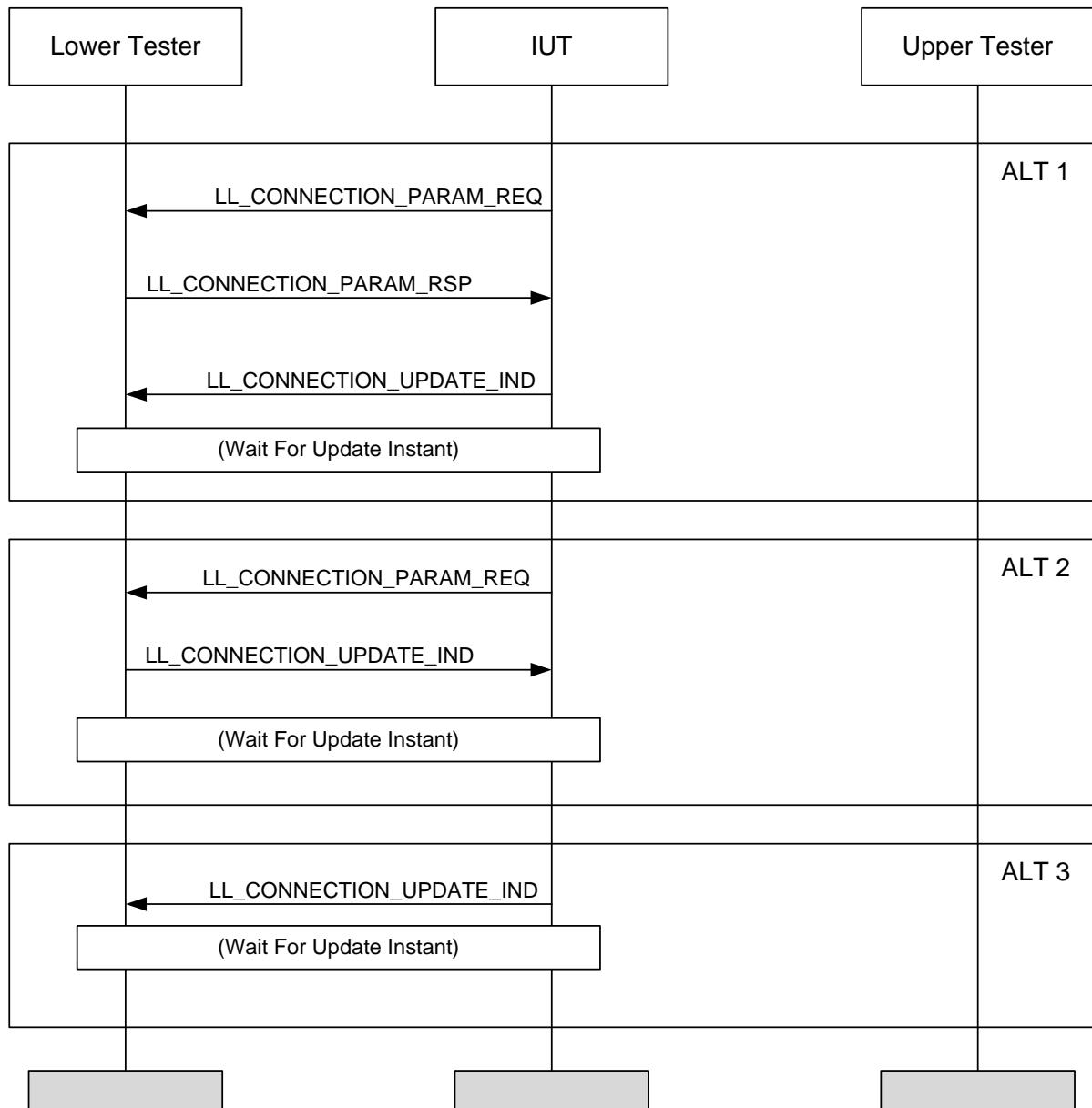


Figure 4.21: Connection Update and Connection Parameters Request Procedure Optional Test Steps

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

- the test does not mention the Connection Update Procedure or the Connection Parameters Request Procedure, or

- the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.
- Channel Map Update Procedure Optional Test Steps

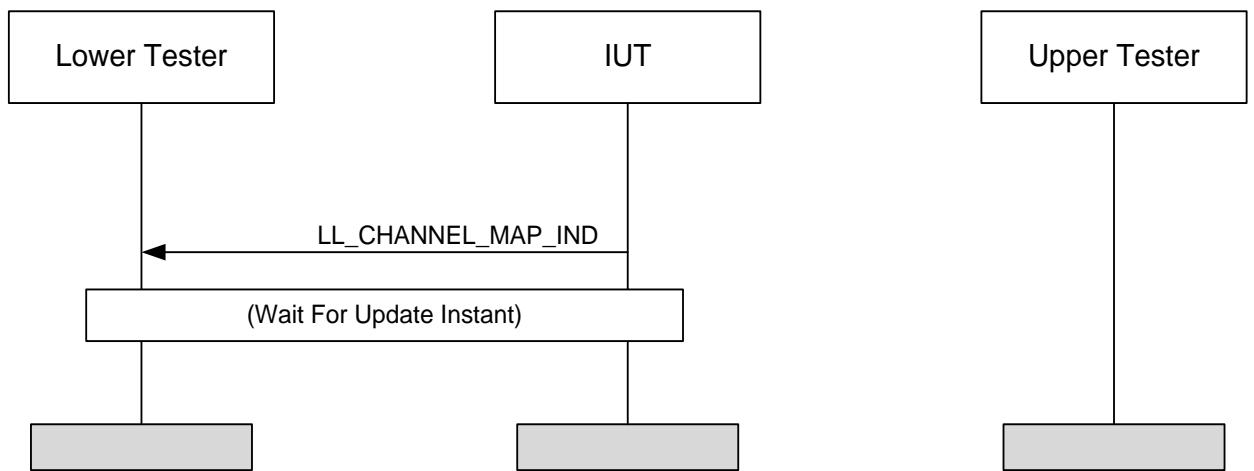


Figure 4.22: Channel Map Update Procedure Optional Test Steps

The IUT may request a channel map update when master only if:

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

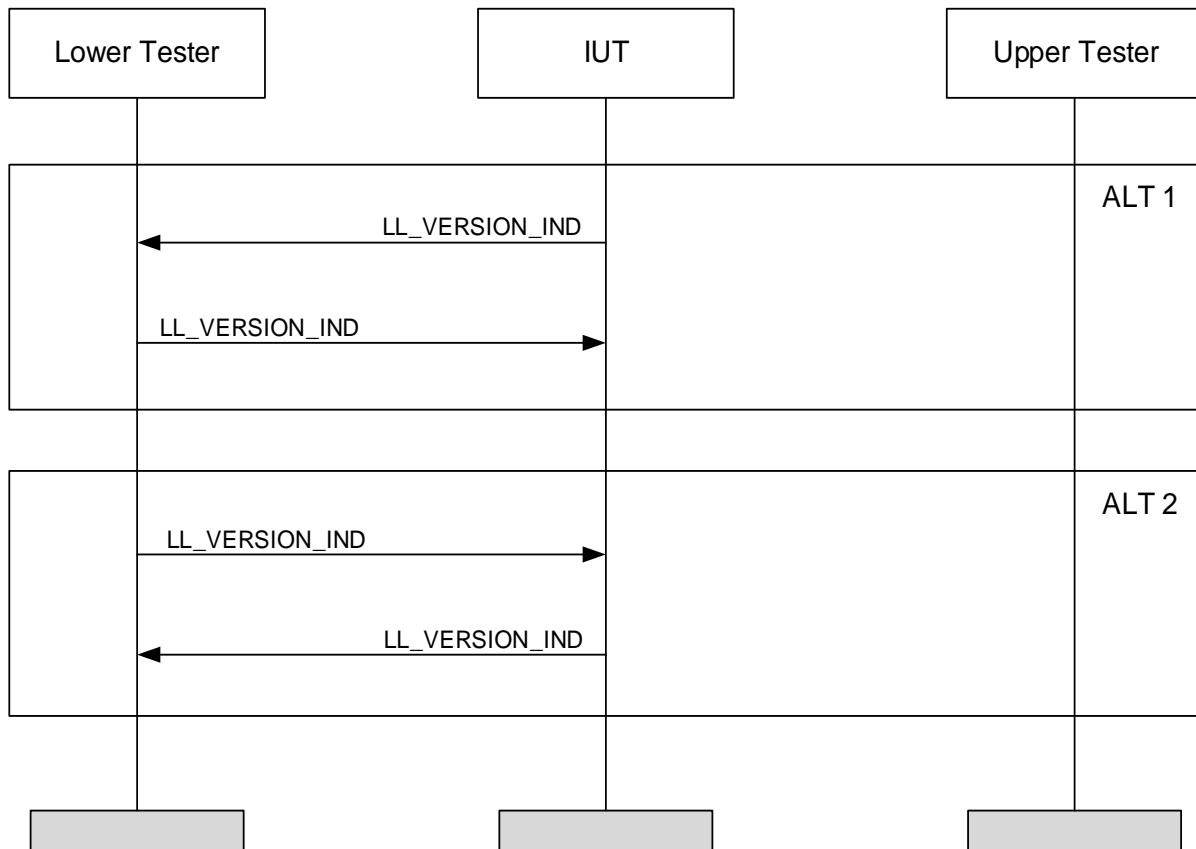


Figure 4.23: Version Exchange Procedure Optional Test Steps

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

The Lower Tester may initiate this procedure to reset/flush any version information that the IUT may have cached from a previous connection with it.

- Feature Exchange Procedure Optional Test Steps

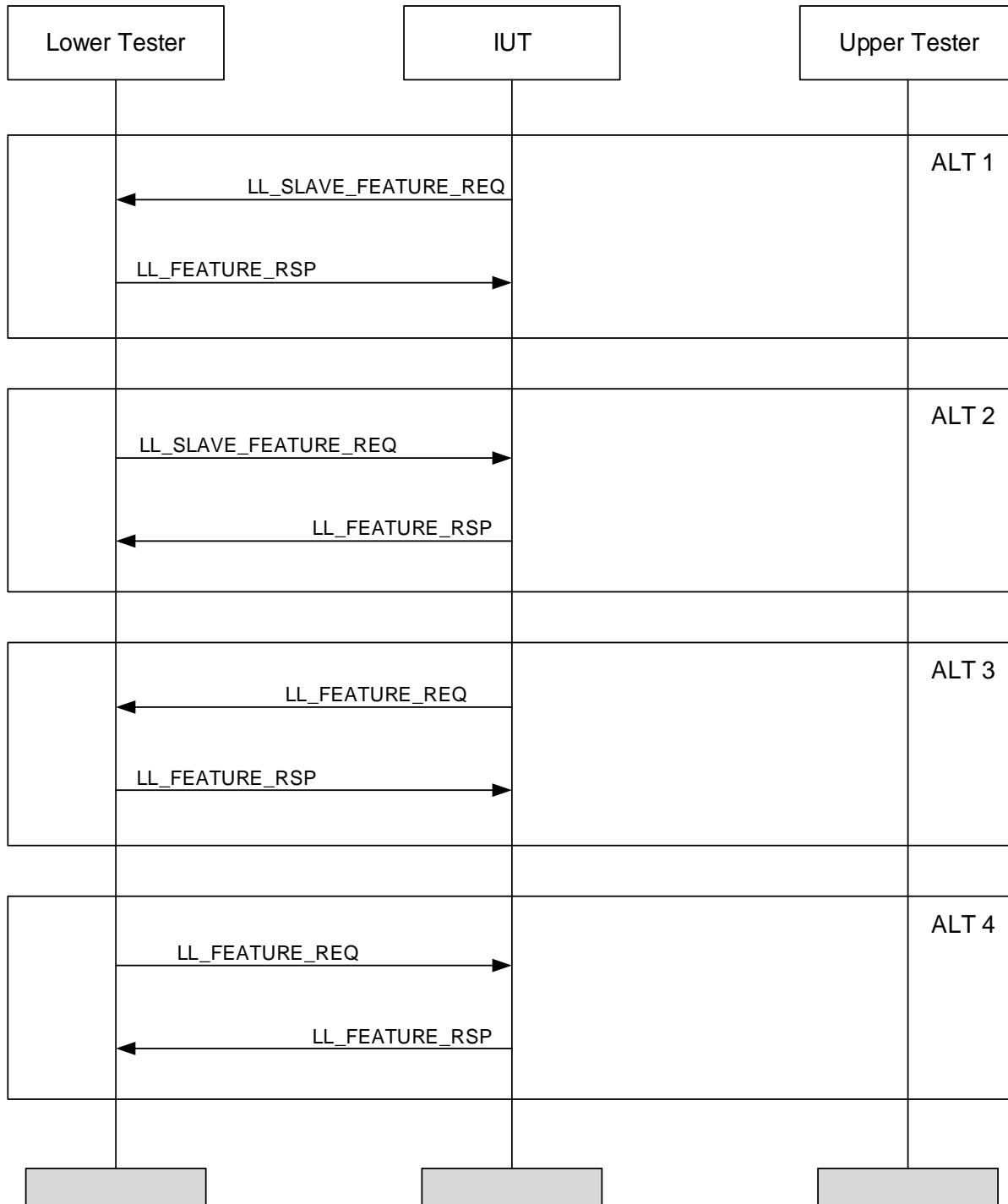


Figure 4.24: Feature Exchange Procedure Optional Test Steps



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

The Lower Tester may initiate this procedure to reset/flush any feature information that the IUT may have cached from a previous connection with it.

- Data Length Update Procedure Optional Test Steps

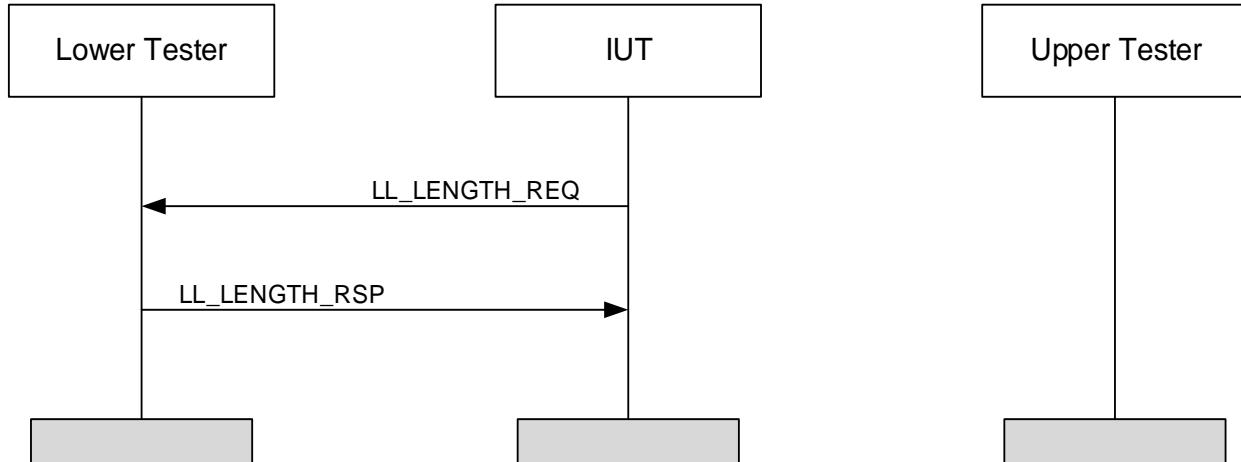


Figure 4.25: Data Length Update Procedure Optional Test Steps

The IUT may initiate a Data Length Update Procedure only if:

- the test does not mention the Data Length Update Procedure, or
- the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.

- LE Ping Procedure Optional Test Steps

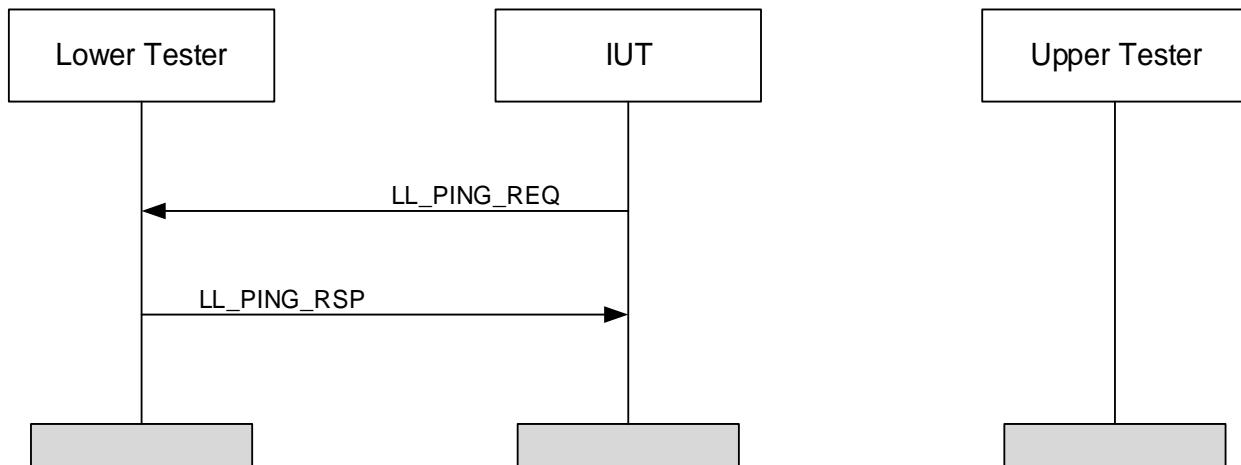


Figure 4.26: LE Ping Procedure Optional Test Steps

IUT may initiate an LE Ping Procedure only if:

- the test does not mention the LE Ping Procedure, or
- the procedure is carried out before the first step of the Test Procedure and the Initial Conditions mentioned in the test are still respected.



- LE PHY Update Procedure Optional Test Steps

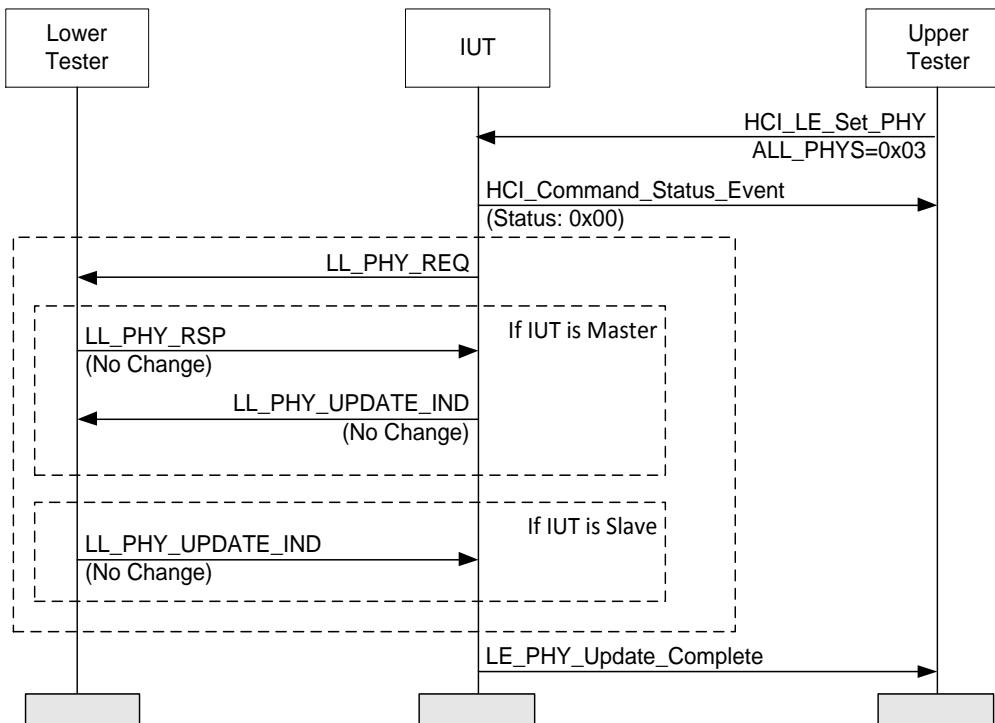


Figure 4.27: LE PHY Update Procedure Optional Test Steps

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

- LE PHY Update Procedure (LE Coded Switch) Optional Test Steps

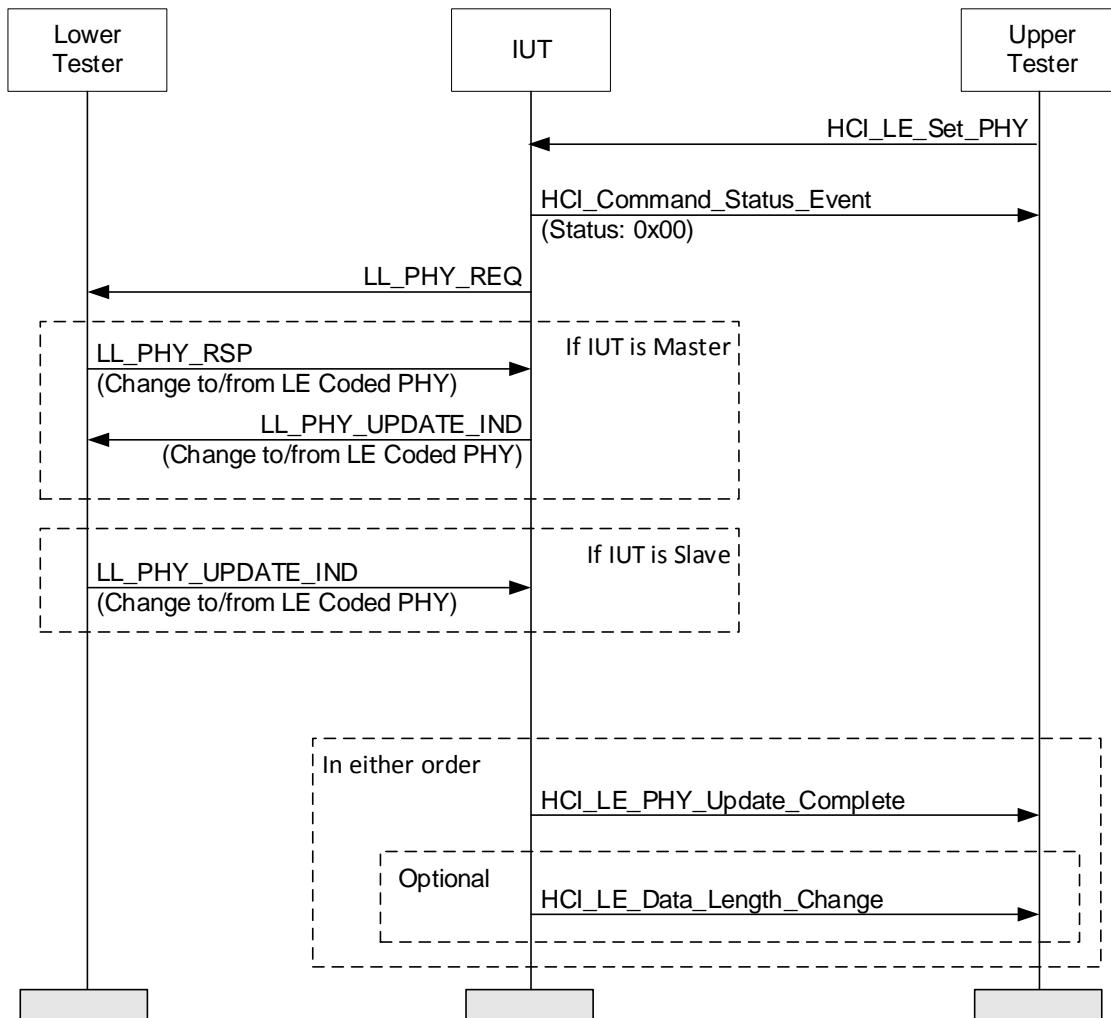


Figure 4.28: LE PHY Update Procedure (LE Coded Switch) Optional Test Steps

If the test procedure involves the Upper Tester issuing an HCI_LE_Set_PHY command that is intended to change the PHY to or from LE Coded PHY, a change to either the maximum Payload length or the maximum transmission time of packets for the IUT may occur. If this is the case, the IUT notifies the Upper Tester with the HCI_LE_Data_Length_Change event immediately before or after the HCI_LE_PHY_Update_Complete event.

4.1.5.8 Pass/Inconclusive/Fail Verdict Conventions

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

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

The convention in this test suite is that, unless there is a specific set of fail conditions outlined in the test case, then the IUT fails the test case as soon one of the pass criteria conditions or the inconclusive



conditions (where they exist) cannot be met, and if this occurs, the outcome of the test shall be the Fail Verdict.

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

4.1.6 Data Fragmentation over HCI

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

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

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

4.1.7 Data PDUs and Empty PDUs

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

4.1.8 Outstanding Commands Prior to Disconnection

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

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

4.2 DDI

Test of the device discovery procedures.

4.2.1 Common PDU Contents

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



4.2.1.1 Legacy Advertising

ADV_NONCONN_IND PDU:

Header						Payload	
lsb msb				lsb msb		LSO MSO lsb msb	lsb msb
Type '0100'	RFU '00'	TxAdd '0'	RFU '0'	Length (6 bits)	RFU '00'	AdvA (6 octets)	AdvData (variable)

ADV_IND PDU:

Header						Payload	
lsb msb					lsb msb	LSO MSO lsb msb	lsb msb
Type '0000'	RFU '0'	ChSel '0'	TxAdd '0'	RFU '0'	Length (8 bits)	AdvA (6 octets)	AdvData (variable)

ADV_DIRECT_IND PDU:

Header						Payload	
lsb msb					lsb msb	LSO MSO lsb msb	LSO MSO lsb msb
Type '1000'	RFU '0'	ChSel '0'	TxAdd '0'	RxAdd '0'	Length '00110000'	AdvA (6 octets)	InitA (6 octets)

ADV_SCAN_IND PDU:

Header						Payload	
lsb msb				lsb msb		LSO MSO lsb msb	lsb msb
Type '0110'	RFU '00'	TxAdd '0'	RFU '0'	Length (6 bits)	RFU '00'	AdvA (6 octets)	AdvData (variable)

SCAN_REQ PDU:

Header						Payload	
lsb msb				lsb msb		LSO MSO lsb msb	LSO MSO lsb msb
Type '1100'	RFU '00'	TxAdd '0'	RxAdd '0'	Length '001100'	RFU '00'	ScanA (6 octets)	AdvA (6 octets)



SCAN_RSP PDU:

Header						Payload	
lsb msb				lsb msb		LSO MSO lsb msb	lsb msb
Type '0010' '00'	RFU '00'	TxAdd (1 bit)	RFU '0'	Length '110100' Or '001100'	RFU '00'	AdvA (6 octets)	ScanRspData (3 to 31 octets)

4.2.1.2 Extended Advertising

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

- If the Upper Tester specified anonymous advertising then neither the ADV_EXT_IND PDU nor the AUX_ADV_IND PDU shall include an AdvA field. Otherwise exactly one shall include an AdvA field. (Note: the Upper Tester will not specify anonymous advertising if no, or zero length, advertising data is specified.)
- If the Upper Tester specified directed advertising then either the ADV_EXT_IND PDU or the AUX_ADV_IND PDU, but not both, shall include a TargetA field. Otherwise neither shall include a TargetA field.
- If an ADV_EXT_IND PDU has an AdvMode field containing 01b (connectable) or 10b (scannable), it shall not include an AdvA or TargetA field.
- If an ADV_EXT_IND PDU is sent on the LE Coded PHY and includes an AuxPtr field, it shall not include an AdvA or TargetA field.
- If a PDU includes an AdvA field, it shall contain the IUT's Advertising Address specified by the Upper Tester during the test procedure.
- If a PDU includes a TargetA field, it shall contain the Lower Tester's Device Address specified by the Upper Tester during the test procedure.

4.2.2 ADV

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

4.2.2.1 Common PDU Contents

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

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

- Test Purpose

Test that an advertiser IUT sends the advertising packets of non-connectable event type, with correct contents and with correct event timing.

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



- Reference
[4] 4.4.2.6, 4.4.2.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

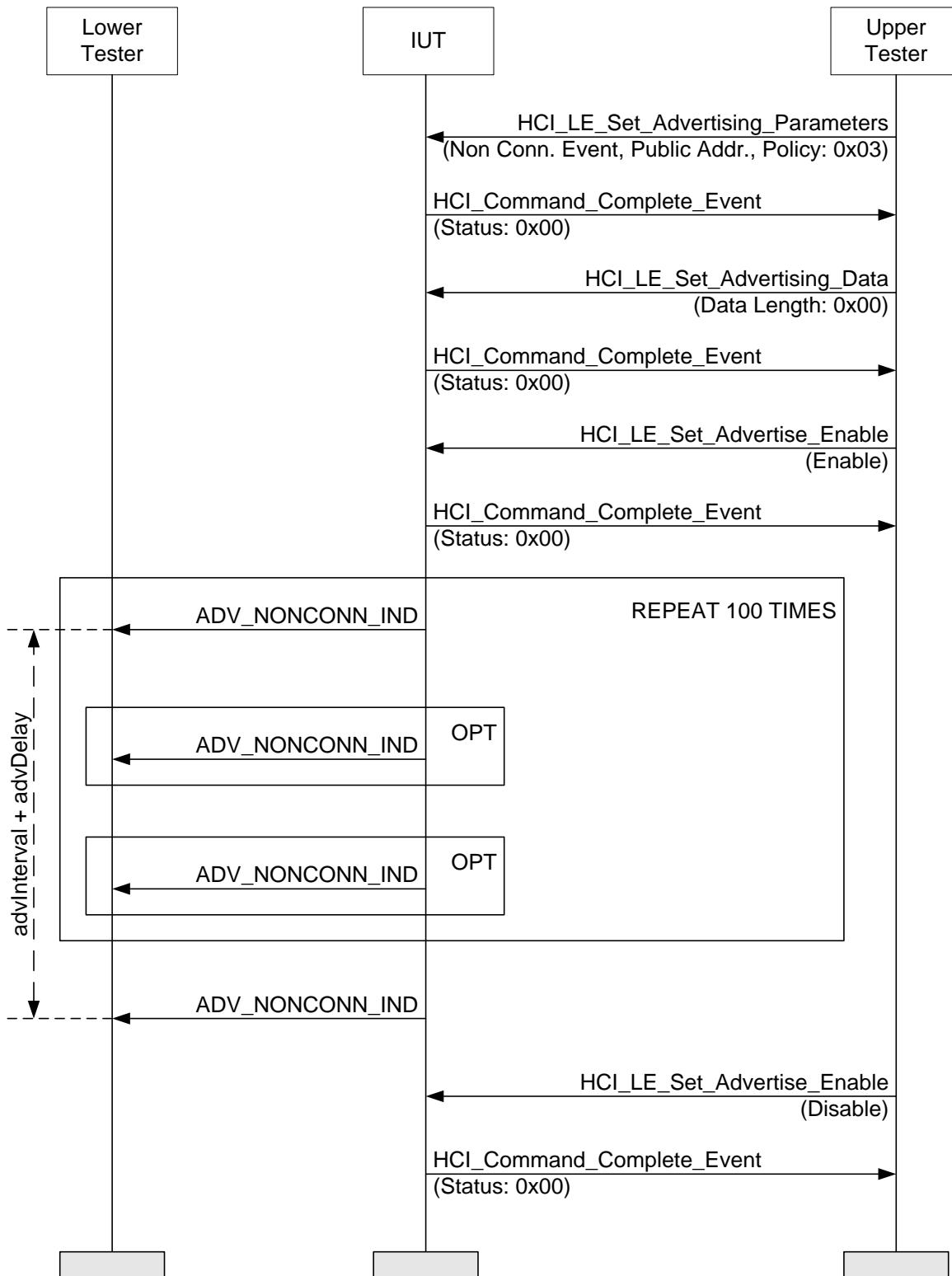


Figure 4.29: LL/DDI/ADV/BV-01-C [Non-Connectable Advertising Events]

- Configure Lower Tester to monitor advertising packets from the IUT.
- Upper Tester enables non-connectable advertising in the IUT using a selected advertising channel and minimum advertising interval.



3. Expect the IUT to send ADV_NONCONN_IND on the selected advertising channel.
 4. Expect the following event to start one advertising interval after the start of the first packet.
 5. Repeat Steps 3-4 until a number of advertising intervals (100) have been detected.
 6. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
- Expected Outcome

Pass Verdict

The test procedure completes using the minimum advertising interval supported for the selected advertising channel.

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

- Notes

The required accuracy for advertising event intervals is a microsecond, from the interval setting granularity of 625 μ s. The delay applied is regarded as a range of up to 10.000 ms, making the timing observation a question of whether the measured range is within the advertising interval plus any delay selected. Taking into account the clock drift of 500 ppm of the low power mode at the minimum advertising interval, tens of microseconds are affected in the accuracy. Jitter for low power mode (16 μ s) has the same effect on the accuracy. The clock drift would affect tens of milliseconds at the maximum advertising interval.

The minimum actual time interval between any two events required is therefore the interval specified plus negative drift plus negative jitter: $(0.625 \text{ ms} * \text{Adv_Interval}) - (0.625 \text{ ms} * \text{Adv_Interval} * 500 \text{ ppm}) - 0.016 \text{ ms}$ from the previous event. The maximum actual time interval is

$(0.625 \text{ ms} * \text{Adv_Interval} + 10.000 \text{ ms}) * (1 + 500 \text{ ppm}) + 0.016 \text{ ms}$ from the previous event.

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

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

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

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

- Test Purpose

Test that an advertiser IUT sends advertising packets of an event with correct contents on a selected advertising channel with correct event timing.

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

- Reference

[3] 4.4.2.3



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

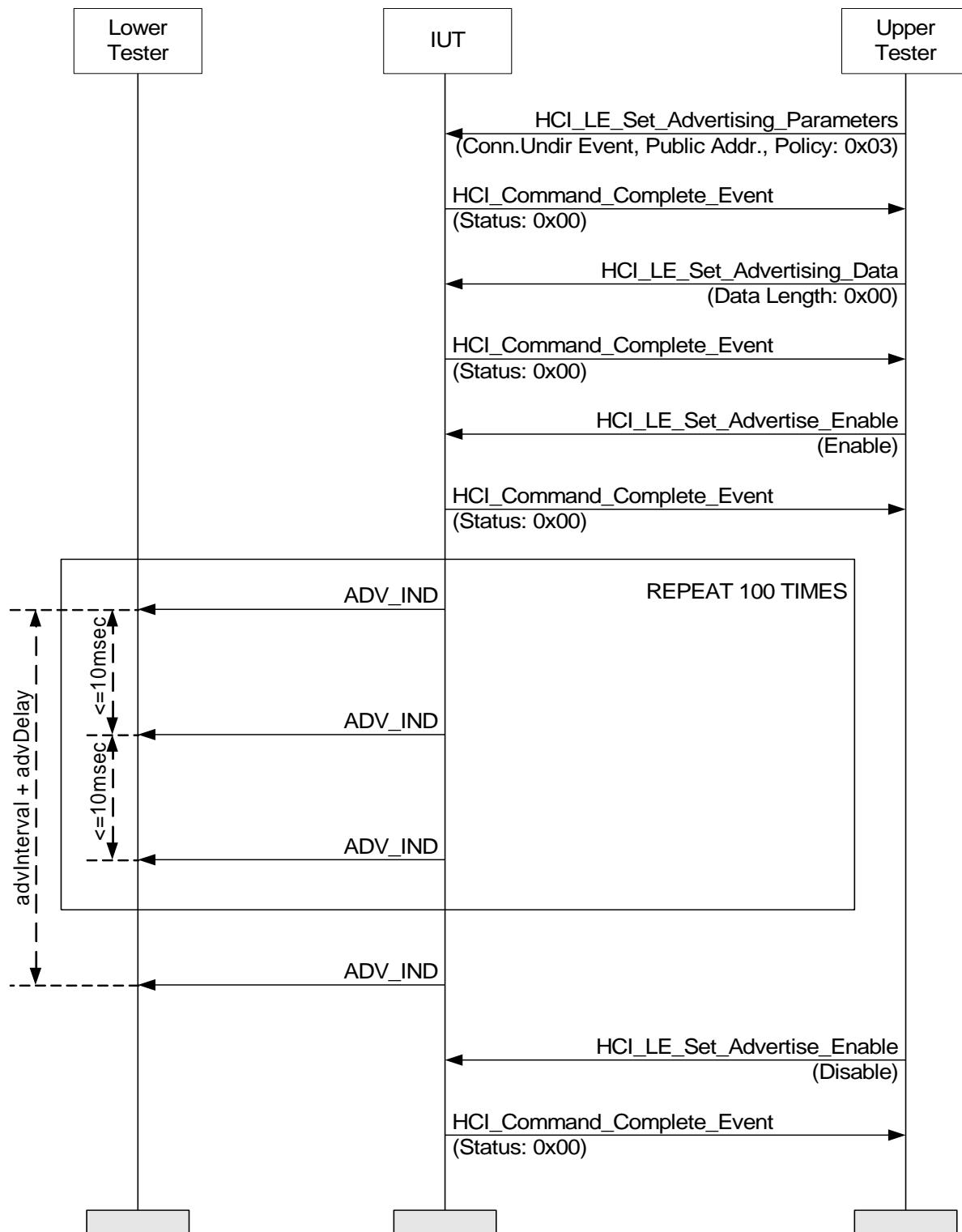


Figure 4.30: LL/DDI/ADV/BV-02-C [Undirected Advertising Events]

- Configure Lower Tester to monitor advertising packets from the IUT.
- Upper Tester enables undirected advertising in the IUT using a selected advertising channel and minimum advertising interval
- Lower Tester expects the IUT to send ADV_IND packets on the selected advertising channel.



4. Expect the next event to start after advertising interval time calculated from the start of the first packet.
 5. Repeat Steps 3-4 until a number advertising intervals (100) have been detected
 6. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
- Expected Outcome

Pass Verdict

The test procedure completes using the minimum advertising interval supported.

The test procedure completes the selected advertising channel.

The timing range detected for advertising events is from 20.0 ms to 30.0 ms (calculated for the minimum advertising interval).
 - Notes

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

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

- Test Purpose

Test that an advertiser IUT sends advertising packets of a non-connectable event type with data on a selected advertising channel.

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

[3] 4.4.2.6
- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using the selected advertising channel, with default data lengths of 1 and 31. If the IUT has a maximum data length of less than 31 bytes, this may be specified via an IXIT value.



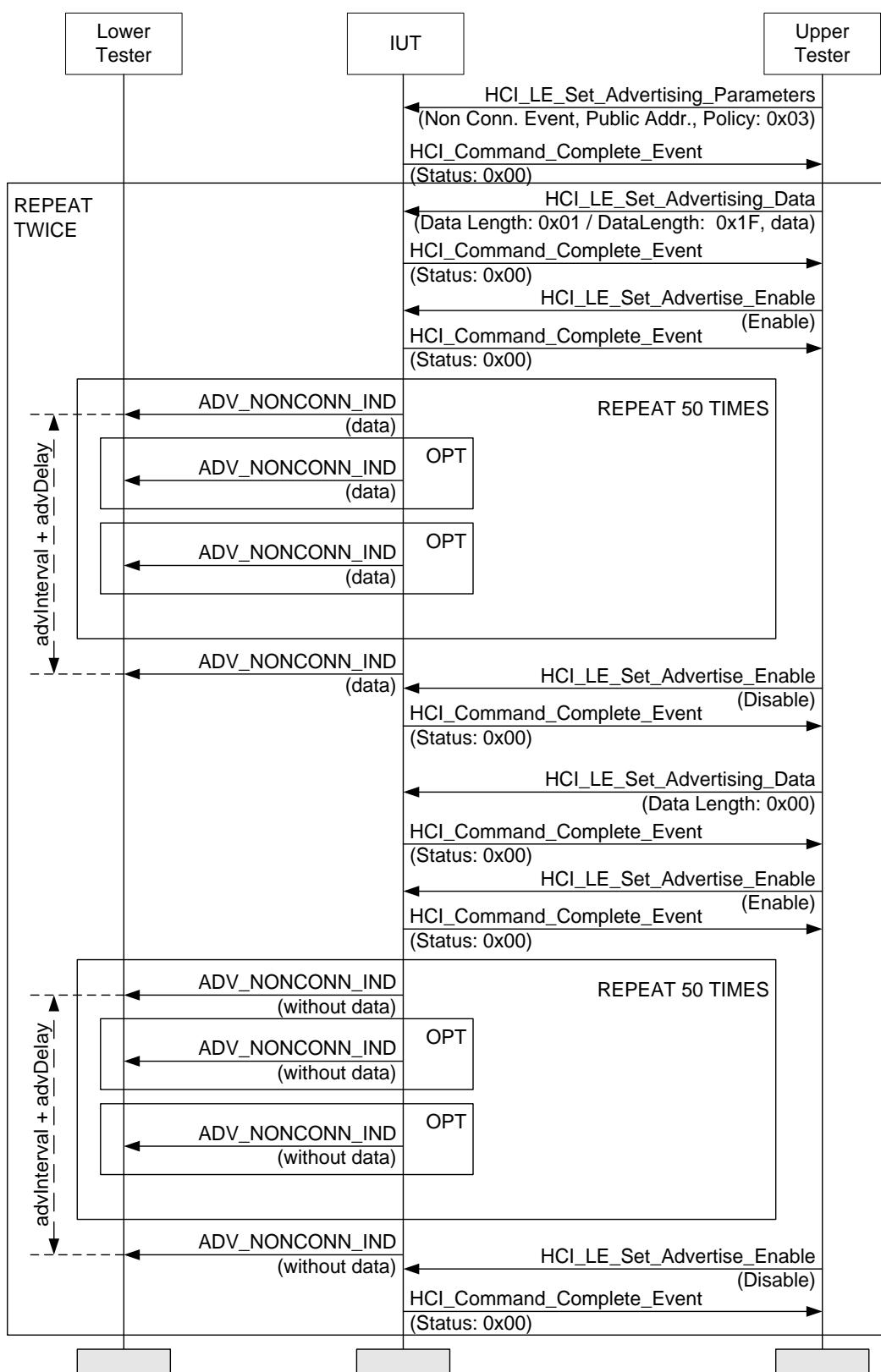


Figure 4.31: LL/DDI/ADV/BV-03-C [Advertising Data: Non-Connectable]



1. Configure Lower Tester to monitor advertising packets from the IUT.
 2. Upper Tester enables non-connectable advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
 3. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and expects an HCI_Command_Complete in response. The data element used in the command is the length of the data field. The data length is 1 byte.
 4. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to enable advertising and expects an HCI_Command_Complete event in response.
 5. Lower Tester expects the IUT to send ADV_NONCONN_IND packets including the data submitted in step 3 starting an event on the selected advertising channel.
 6. Expect the following event to start after advertising interval time calculating from the start of the first packet.
 7. Repeat Steps 5-6 until a number of advertising intervals (50) have been detected
 8. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising function and expects an HCI_Command_Complete event in response.
 9. Upper Tester sends an HCI_LE_Set_Advertising_Data to configure the IUT to send advertising packets without advertising data and expects an HCI_Command_Complete event in response.
 10. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to enable advertising and expects an HCI_Command_Complete event in response.
 11. Lower Tester expects the IUT to send ADV_NONCONN_IND packets including no advertising data starting an event on the selected advertising channel.
 12. Expect the next event to start after advertising interval time calculating from the start of the first packet.
 13. Repeat Steps 11-12 until a number of advertising intervals (50) have been detected
 14. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.
 15. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and expects an HCI_Command_Complete in response. The data element is a number indicating the length of the data field in the first octet encoded unsigned least significant bit first and the rest of the octets zeroes. The data length is either 31 bytes by default or it may be specified by IXIT value if less than 31 bytes.
 16. Repeat steps 4 to 14.
- Expected Outcome

Pass Verdict

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

The IUT transmits data as submitted in the HCI commands.

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

- Test Purpose

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

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

- Reference

[3] 4.4.2.3



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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



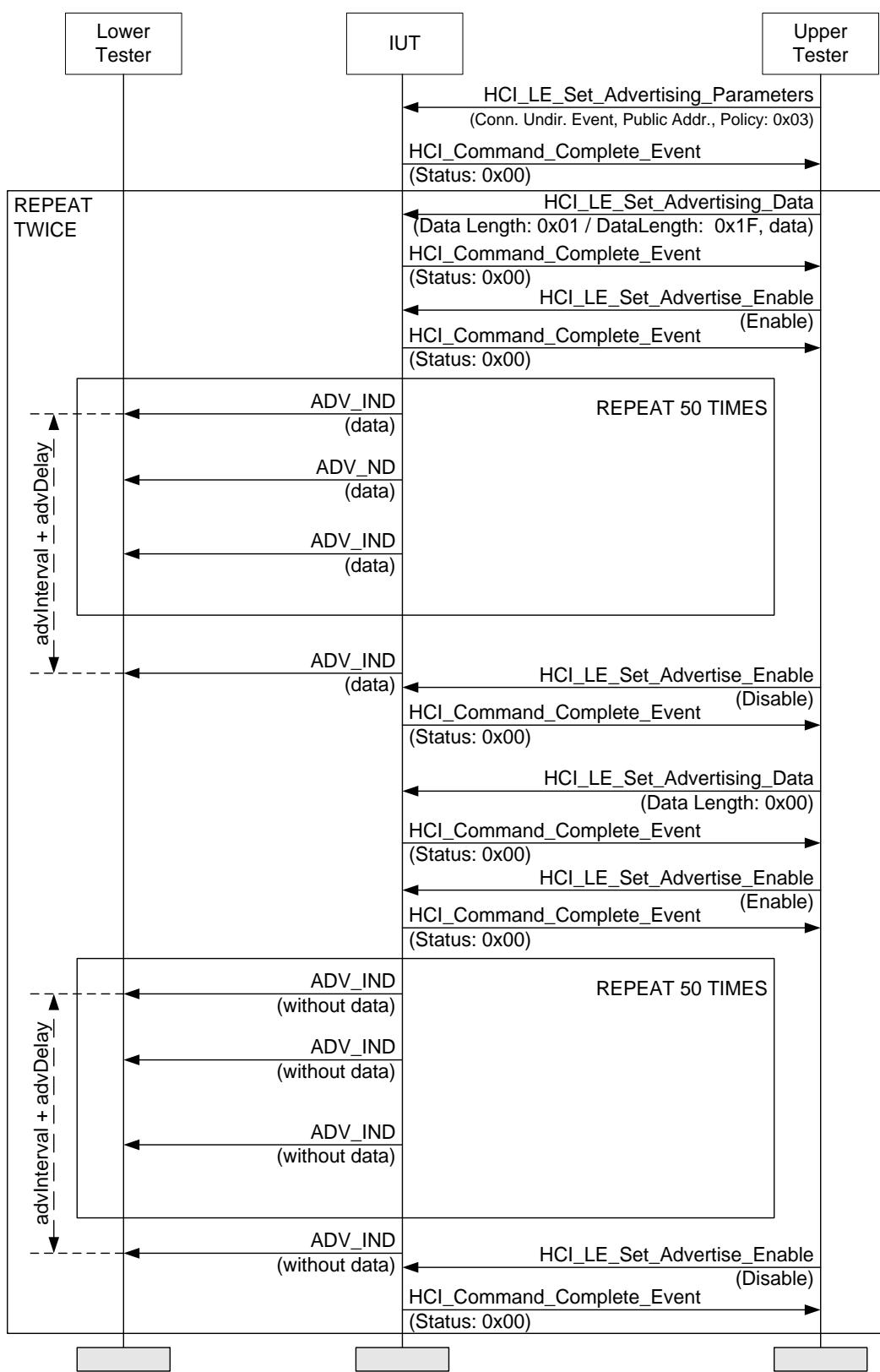


Figure 4.32: LL/DDI/ADV/BV-04-C [Advertising Data: Undirected]

1. Configure Lower Tester to monitor advertising packets from the IUT.
2. Upper Tester enables undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.



3. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and expects an HCI_Command_Complete in response. The data element used in the command is the length of the data field. The data length is 1 byte.
 4. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to enable advertising and expects an HCI_Command_Complete event in response.
 5. Lower Tester expects the IUT to send ADV_IND packets including the data submitted in step 3 starting an event on the selected advertising channel.
 6. Expect the following event to start after advertising interval time calculating from the start of the first packet.
 7. Repeat Steps 5-6 until a number of advertising intervals (50) have been detected
 8. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising function and expects an HCI_Command_Complete event in response.
 9. Upper Tester sends an HCI_LE_Set_Advertising_Data to configure the IUT to send advertising packets without advertising data and expects an HCI_Command_Complete event in response.
 10. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to enable advertising and expects an HCI_Command_Complete event in response.
 11. Lower Tester expects the IUT to send ADV_IND packets including no advertising data starting an event on the selected advertising channel.
 12. Expect the next event to start after advertising interval time calculating from the start of the first packet.
 13. Repeat Steps 11-12 until a number of advertising intervals (50) have been detected
 14. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.
 15. Upper Tester sends an HCI_LE_Set_Advertising_Data command to the IUT and expects an HCI_Command_Complete in response. The data element is a number indicating the length of the data field in the first octet encoded unsigned least significant bit first and the rest of the octets zeroes. The data length is 31 bytes.
 16. Repeat steps 4 to 14.
- Expected Outcome

Pass Verdict

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

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

- Test Purpose

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

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

- Reference

[3] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map.

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



name) AND White List All Unknown Devices (Allow Scan Request from Any, Allow Connect Request from Any (0x00)).

- Test Procedure

Execute the test procedure using the minimum advertising interval supported.

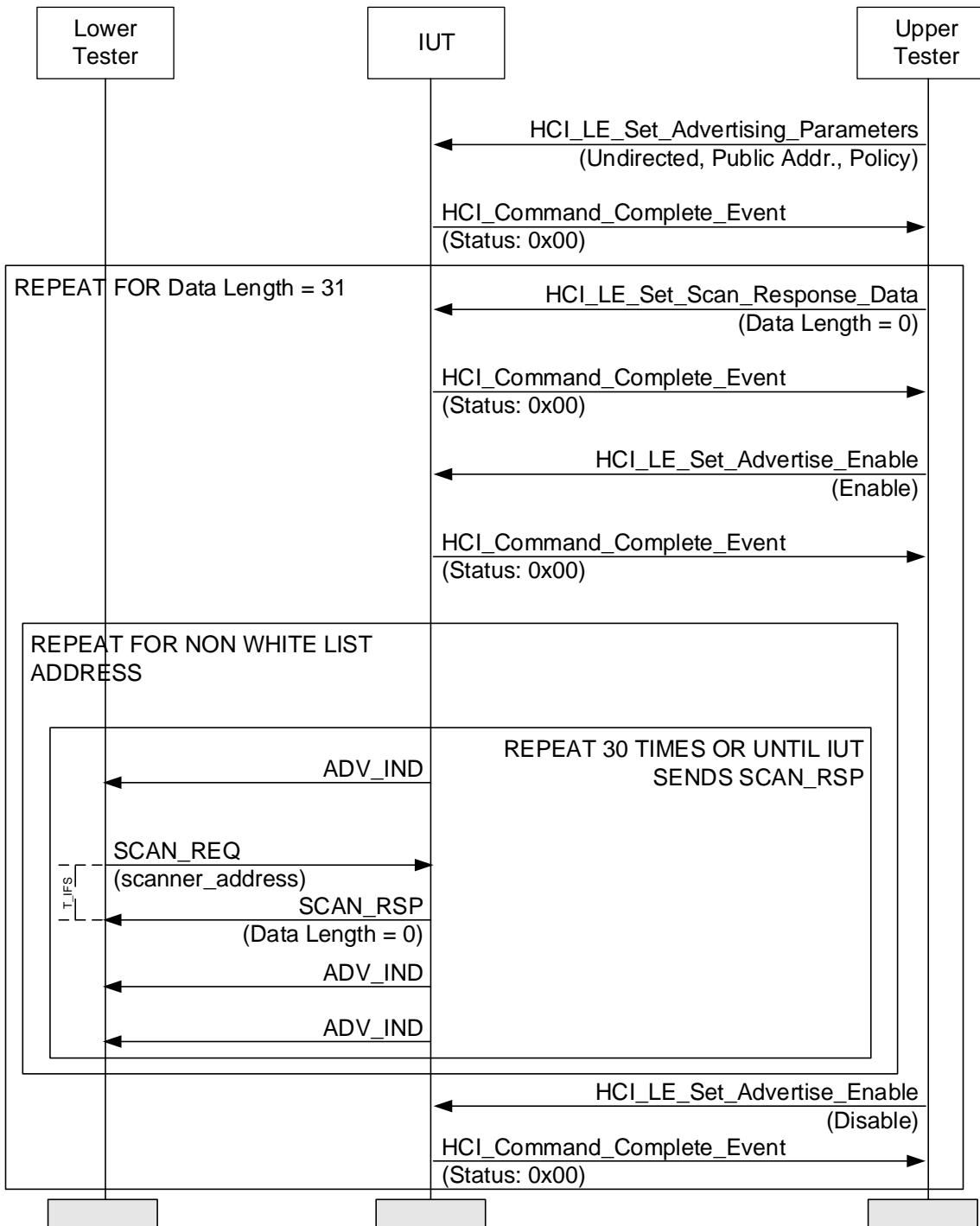


Figure 4.33: LL/DDI/ADV/BV-05-C [Scan Request: Undirected Connectable]



1. Upper Tester enables undirected advertising in the IUT using all supported advertising channels and minimum advertising intervals.
 2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a common device address as parameter.
 3. Configure Scan Response Data in the IUT using device name length of 0 as response data.
 4. Lower Tester sends a SCAN_REQ packet on the selected advertising channel after receiving an ADV_IND packet from IUT on the advertising channel configured in step 3. The SCAN_REQ is sent TIFS after the end of an ADV_IND packet.
 5. Lower Tester expects a SCAN_RSP packet from the IUT addressed to the Lower Tester TIFS after the end of the request packet.
 6. Repeat steps 4-5 30 times or until IUT sends a SCN_RSP.
 7. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 8. Repeat steps 4-6.
 9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a device address that has an identical company assigned identifier to the address of the IUT but a different company identifier as parameter.
 10. Configure Scan Response Data in the IUT using device name length of 0 as response data.
 11. Repeat steps 4-6.
 12. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 13. Repeat steps 4-6.
 14. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a device address that has an address mismatching the IUT address in both the company assigned identifier and the company identifier as parameter. Configure Scan Response Data in the IUT using device name length of 0 as response data.
 15. Repeat steps 4-6.
 16. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 17. Repeat steps 4-6.
- Expected Outcome
- Pass Verdict
- The test procedure completes using the minimum advertising interval supported,
- The test procedure completes with the IUT responding on each advertising channel,
- The IUT responds in each case of different scanner address used,
- The timing deviations detected for packets in active mode are within the 2 μ s range around TIFS.
- 4.2.2.7 LL/DDI/ADV/BV-06-C [Connection Request]**
- Test Purpose
- Tests that an advertiser IUT receives a connection request and stops advertising after its reception.
- The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.
- Reference
- [3] 4.4.2.3, 4.3.2



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

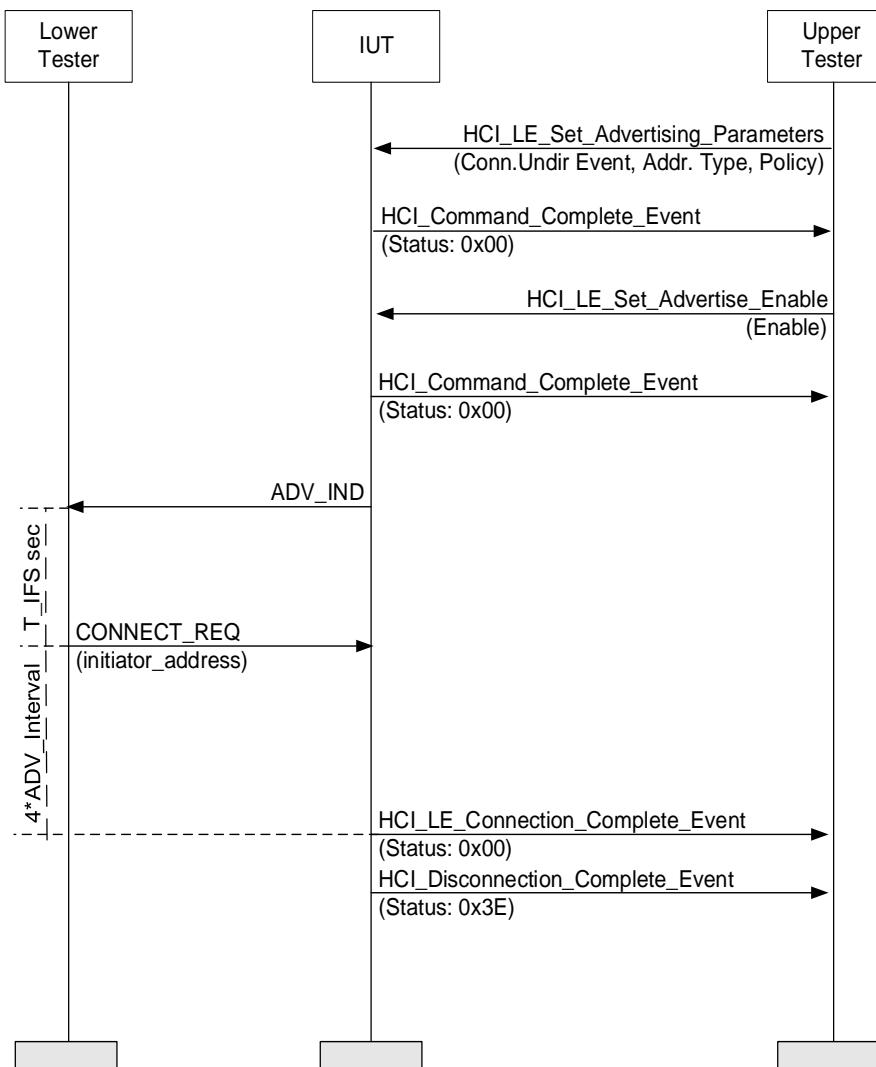


Figure 4.34: LL/DDI/ADV/BV-06-C [Connection Request]

- Upper Tester enables undirected advertising in the IUT using all supported advertising channels and minimum advertising intervals.
- Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a `CONNECT_IND` packet on the first supported advertising channel.
- Configure Lower Tester to use a white device address as parameter of `CONNECT_IND`.
- The Lower Tester expects an `ADV_IND` packet from the IUT and responds with a `CONNECT_IND` packet T_{IFS} after the end of the advertising packet.



5. The Lower Tester expects no ADV_IND packet after the advertising interval from the IUT. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 4 and as postamble: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
 7. Configure Lower Tester to use a device address with an identical company assigned identifier but a different company identifier than the IUT) as parameter of CONNECT_IND.
 8. Repeat steps 4-6.
 9. Configure Lower Tester to use a device address with a different company assigned identifier and the same company identifier as the IUT as parameter of CONNECT_IND.
 10. Repeat steps 4-6.
 11. Configure Lower Tester to use a device address mismatching the IUT address in both the company assigned identifier and the company identifier as parameter of CONNECT_IND.
 12. Repeat steps 4-6.
 13. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the second supported advertising channel.
 14. Repeat steps 3-12.
 15. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the third supported advertising channel.
 16. Repeat steps 3-12.
- Expected Outcome

Pass Verdict

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

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

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

- Test Purpose

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

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

- Reference

[3] 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

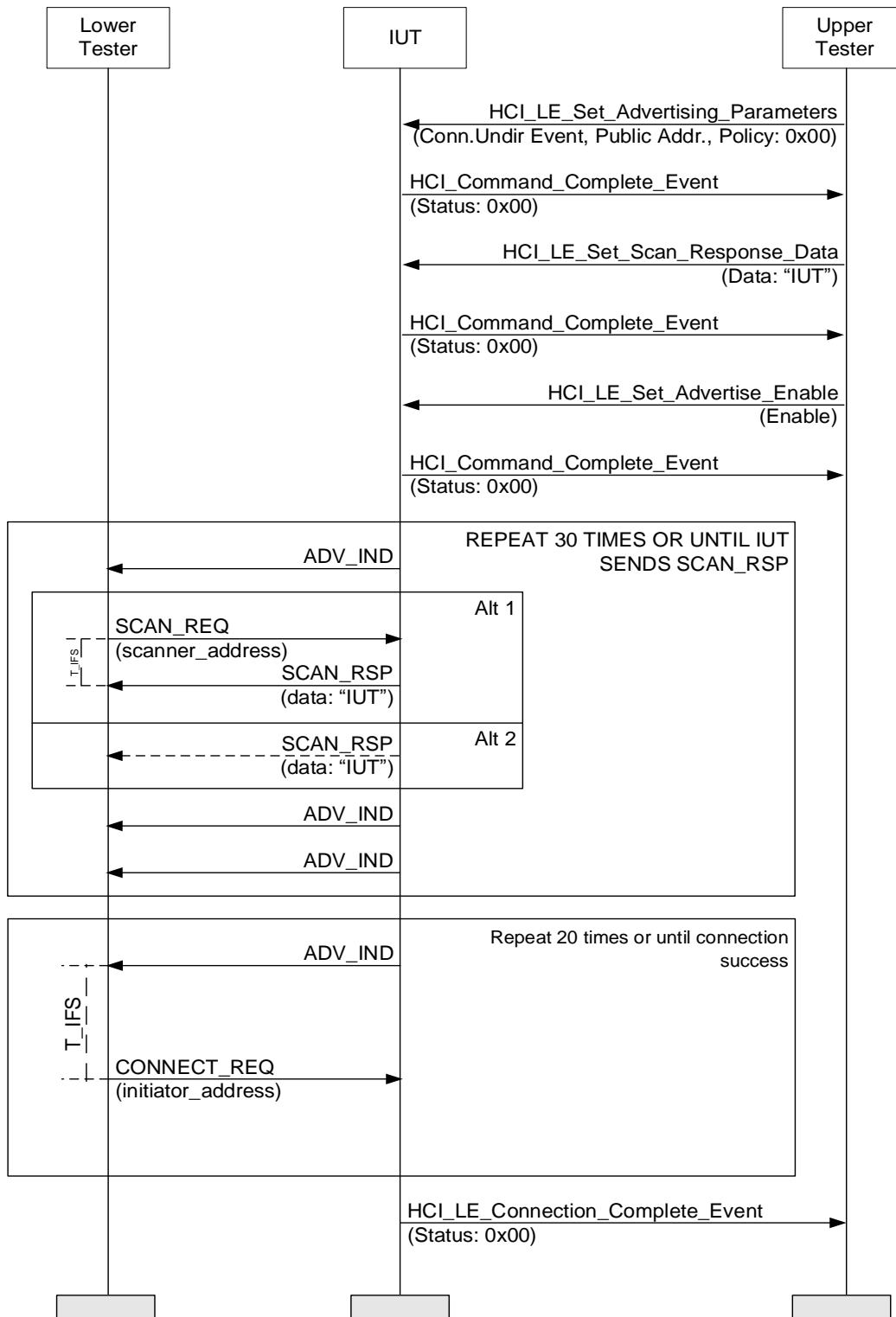


Figure 4.35: LL/DDI/ADV/BV-07-C [Scan Request Connection Request]

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



2. Upper Tester sends an HCI_LE_Set_Scan_Response_Data command with data set to "IUT" and expects an HCI_Command_Complete event from the IUT.
 3. Configure Lower Tester to monitor the advertising, scan response and connection procedures of the IUT, sending a SCAN_REQ and a CONNECT_IND packet on a supported advertising channel (defined as an IXIT).
 4. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with an SCAN_REQ packet on the selected advertising channel T_IFS after the end of an advertising packet.
 5. Lower Tester expects an SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
 6. Repeat steps 4-5 30 times or until IUT sends SCAN_RSP.
 7. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
 8. The Lower Tester expects no ADV_IND packet after advertising interval from the IUT after sending the connection request to indicate that the IUT has stopped advertising.
 9. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT.
 10. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 9).
- Expected Outcome

Pass Verdict

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

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

- Test Purpose

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

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

- Reference

[3] 4.3.2, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

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



- Test Procedure

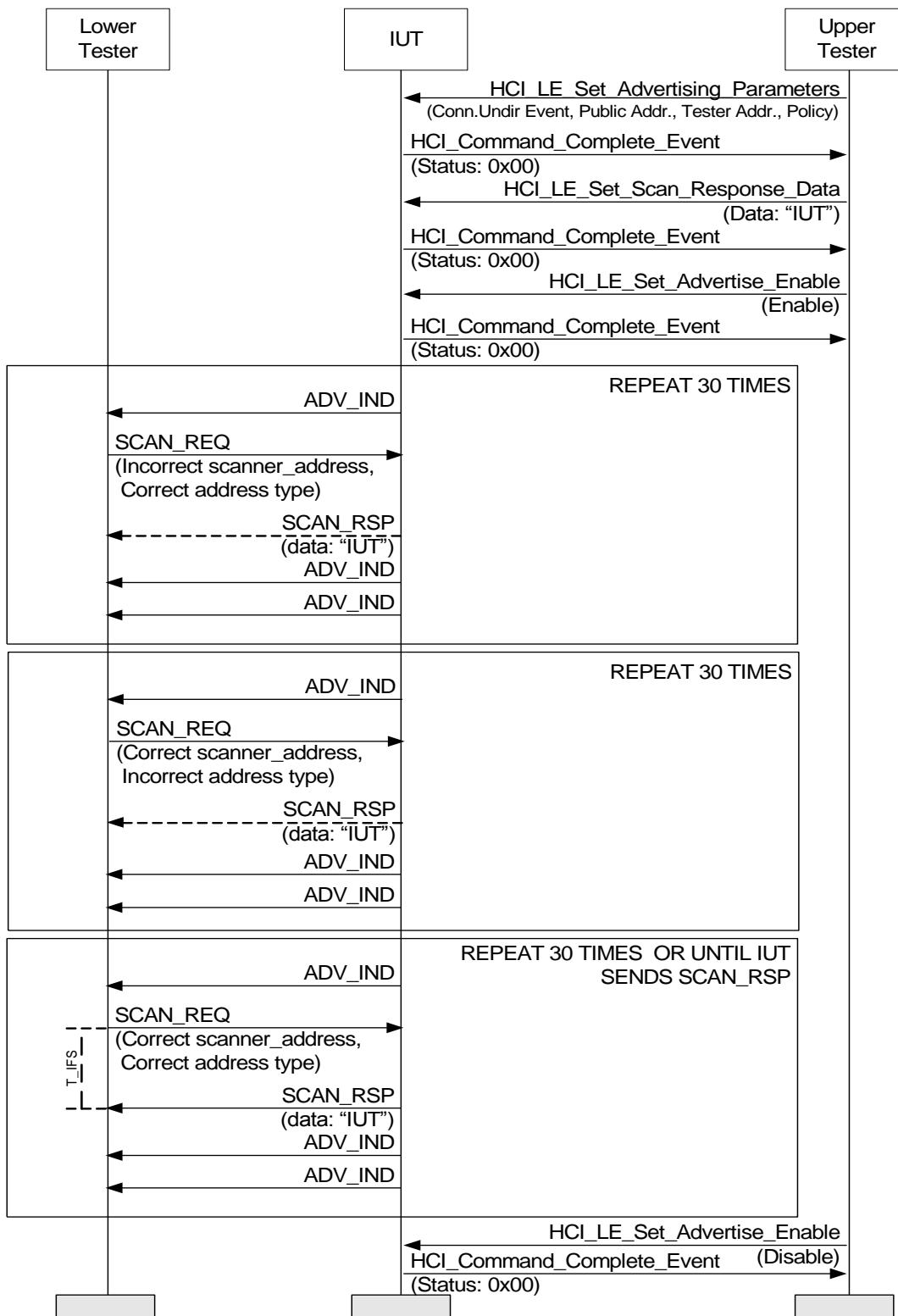


Figure 4.36: LL/DDI/ADV/BV-08-C [Scan Request Device filtering]

- Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising



- intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.
2. Upper Tester sends an HCI_LE_Set_Scan_Response_Data command with data set to “IUT” and expects an HCI_Command_Complete event from the IUT.
 3. Lower Tester address type is set to Public Address type.
 4. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send an SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address differing in the company assigned identifier (an address black listed in the policy applied) and correct address type.
 5. Lower Tester expects an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with the selected address on the selected advertising channel T_IFS after the end of an advertising packet.
 6. Lower Tester expects no response from the IUT.
 7. Repeat steps 5-6 30 times.
 8. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and an incorrect address type.
 9. Repeat steps 5-6 30 times.
 10. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and correct address type.
 11. Lower Tester expects an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with an address white listed in the policy applied using correct address type, on the selected advertising channel T_IFS after the end of an advertising packet.
 12. Lower Tester expects a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
 13. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.
 14. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from White List (0x03)’.
 15. Lower Tester address type is set to Random Address type.
 16. Repeat steps 4-13.
 17. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from Any (0x01)’.
 18. Lower Tester address type is set to Public Address type.
 19. Repeat steps 4-13.
 20. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to ‘Allow Scan Request from White List, Allow Connect Request from Any (0x01)’.
 21. Lower Tester address type is set to Random Address type.
 22. Repeat steps 4-13.



- Expected Outcome

Pass Verdict

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

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

The IUT does not respond to the advertising packets with the white listed addresses,

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

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

- Test Purpose

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

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

- Reference

[3] 4.3.2, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

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



- Test Procedure

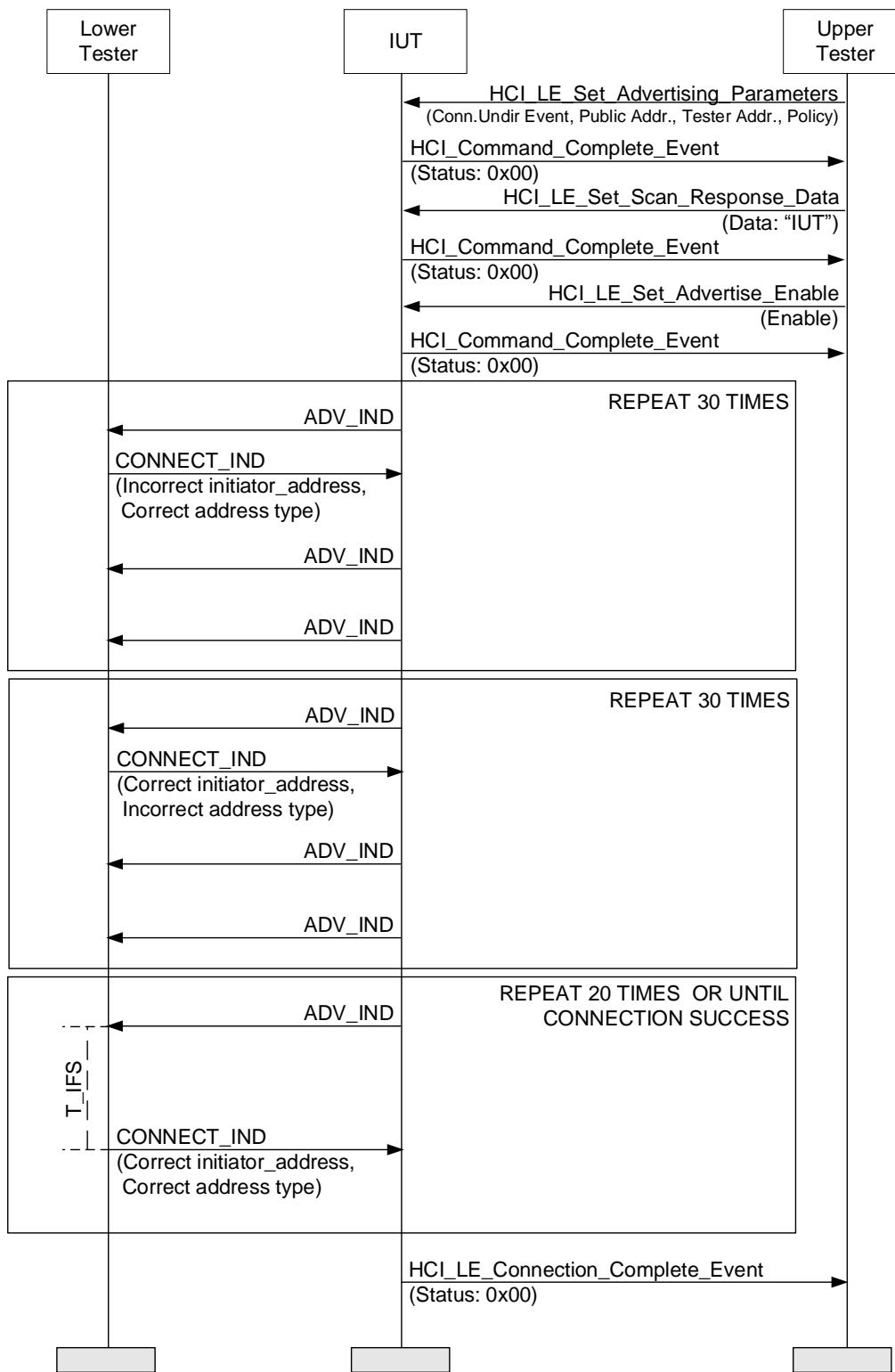


Figure 4.37: LL/DDI/ADV/BV-09-C [Connection Request Device Filtering]

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

2. Upper Tester sends an HCI_LE_Set_Scan_Response_Data command with data set to "IUT" and expects an HCI_Command_Complete event from the IUT.
3. Lower Tester address type is set to Public Address type.
4. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the selected supported advertising channel (defined as an IXIT) in response to connectable advertisements. The initiator's address in the CONNECT_IND PDU shall be formed by using the same address type as the entry on the IUT's white list but changing the company identifier within the address to ensure a mis-match.
5. Lower Tester expects an ADV_IND packet from the IUT and responds with a CONNECT_IND packet with the selected address on the selected advertising channel T_IFS after the end of an advertising packet.
6. Lower Tester expects the IUT to continue advertising.
7. Repeat steps 5 and 6 30 times.
8. Configure Lower Tester to use a device address on the IUT's white list but an incorrect address type as the address parameter of the CONNECT_IND PDU.
9. Repeat steps 5 and 6 30 times.
10. Configure Lower Tester to use a device address on the IUT's white list and correct address type as the address parameter of the CONNECT_IND PDU.
11. Lower Tester expects an ADV_IND packet from the IUT and responds with a CONNECT_IND packet with the selected address white listed in the policy applied, on the selected advertising channel T_IFS after the end of an advertising packet.
12. The Lower Tester expects no ADV_IND packet after advertising interval from the IUT after sending the connection request to indicate that the IUT has stopped advertising. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.
13. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT.
14. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
15. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from White List (0x03)'.
16. Lower Tester address type is set to Random Address type.
17. Repeat steps 4-14.
18. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to 'Allow Scan Request from Any, Allow Connect Request from white list (0x02)'.
19. Lower Tester address type is set to Public Address type.
20. Repeat steps 4-14.
21. Upper Tester enables undirected advertising in the IUT using public address type, all supported advertising channels and filtering policy set to 'Allow Scan Request from Any, Allow Connect Request from white list (0x02)'.
22. Lower Tester address type is set to Random Address type.
23. Repeat steps 4-14.
24. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising interval and filtering policy set to 'Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)'.
25. Configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the first supported advertising channel in response to connectable advertisements. The initiator's address in the CONNECT_IND PDU shall be an address on the IUT's white list.



26. Lower Tester expects an ADV_IND packet from the IUT and responds with a CONNECT_IND packet TIFS after the end of the advertising packet.
 27. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's corrected formatted LL Data Packets on the data channels. If no data packets are received, repeat steps 26 and 27 up to 20 times or until the IUT stops advertising.
 28. The Lower Tester expects no ADV_IND packet after advertising interval from the IUT after sending the connection request. Wait for a time equal to 4 advertising intervals to check that no ADV_IND is received.
 29. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 25 and as postamble: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
 30. Repeat steps 24 to 29, except that in Step 25, configure Lower Tester to use a device address not on the IUT's white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT's white list but changing the company identifier within the address to ensure a mis-match.
 31. Repeat steps 24 to 29, except that in Step 25, configure Lower Tester to use a device address not on the IUT's white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT's white list, leaving the company identifier within the address correct, but changing the company-assigned identifier to ensure a mis-match.
 32. Repeat steps 24 to 29, except that in Step 25, configure Lower Tester to use a device address not on the IUT's white list as the address parameter of the CONNECT_IND PDU; the address shall be formed by using the same address type as the entry on the IUT's white list but changing both the company identifier and company-assigned identifier to ensure a mis-match.
 33. Repeat steps 24-32, except that in Step 25, configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the second supported advertising channel in response to connectable advertisements.
 34. Repeat steps 24-32, except that in Step 25, configure Lower Tester to monitor the advertising and connection procedures of the IUT and send a CONNECT_IND packet on the third supported advertising channel in response to connectable advertisements.
 35. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising intervals and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from Any (0x01)'.
 36. Repeat steps 25-29.
- Expected Outcome

Pass Verdict

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

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

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

- Test Purpose

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



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

- Reference

[3] 4.4.2.4

- Initial Condition

Parameters: LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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



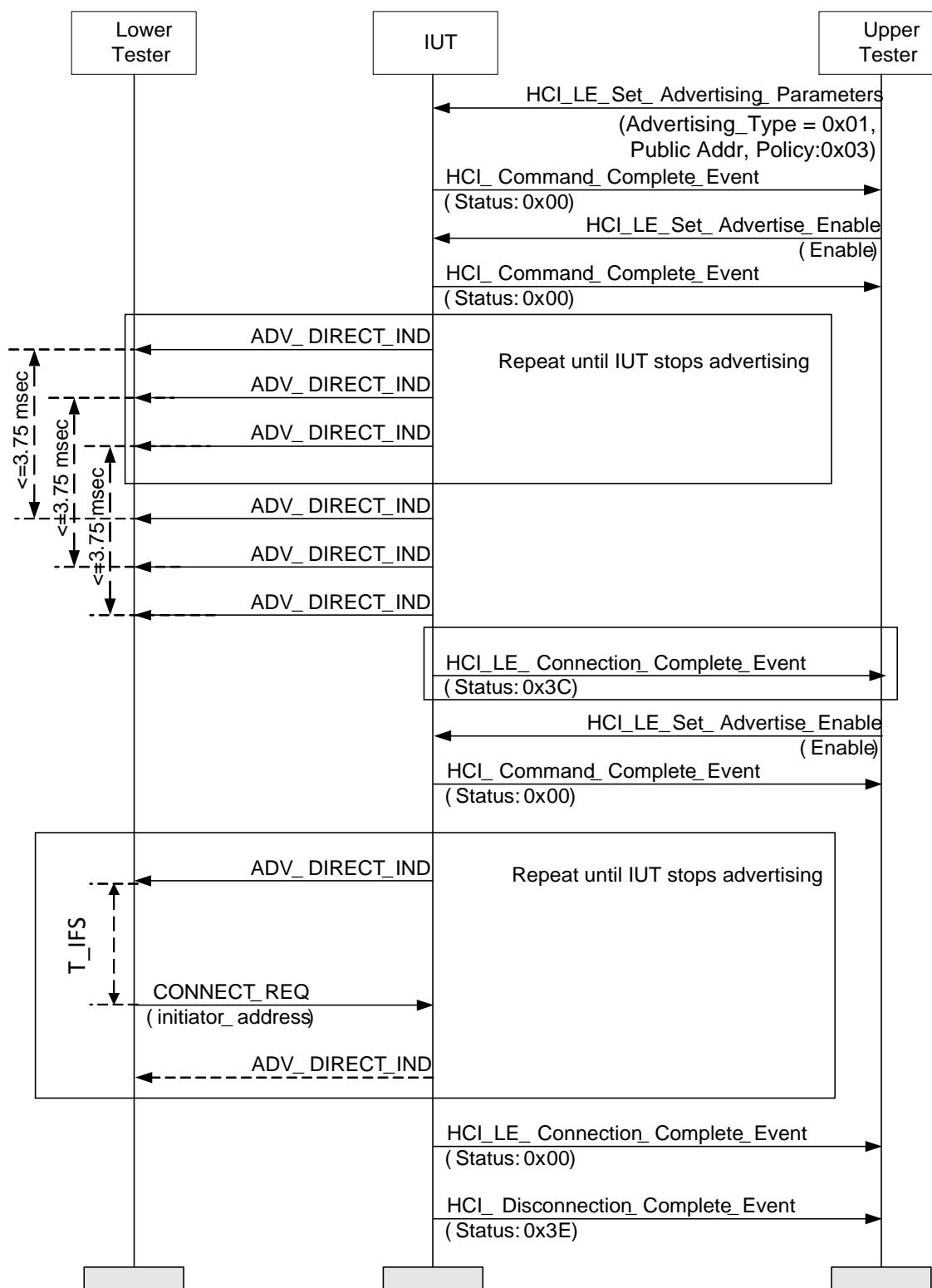


Figure 4.38: LL/DDI/ADV/BV-11-C [Directed Advertising Events]

1. Configure Lower Tester to start passive scanning.
2. Upper Tester enables high duty cycle directed advertising in the IUT using all supported advertising channels.
3. Lower Tester expects the IUT to send ADV_DIRECT_IND packets: A packet starting an event on an applicable advertising channel with the lowest advertising channel index, then optionally



following packets on applicable advertising channels with increasing advertising channel indexes. Expect the intervals between starts of packet on any single channel to be equal to or below 3.75 ms.

4. Repeat until the IUT stops advertising and verify that it stops after 1.28s. For each advertising channel, verify that at least 30 of the intervals between starts of packets on that channel are equal to or below 3.75 ms.
 5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT with status parameter set to 'directed advertising timeout'.
 6. Configure Lower Tester to initiate a connection.
 7. Upper Tester enables directed advertising in the IUT using all supported advertising channels.
 8. Lower Tester expects an ADV_DIRECT_IND packet from the IUT on the selected advertising channel (defined as an IXIT), then responds with a CONNECT_IND packet T_IFS after the end of the advertising packet and does not send any data packets to the IUT.
 9. Lower Tester expects no ADV_DIRECT_IND packets from the IUT after the advertising interval.
 10. Repeat steps 8-9 until the IUT stops advertising.
 11. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 8.
 12. Upper Tester expects an HCI_Disconnection_Complete event from the IUT with the reason parameter indicating 'connection failed to be established', with the connection handle parameter matching to Step 8.
- Expected Outcome

Pass Verdict

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

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

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

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

- Notes

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

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

- Test Purpose

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

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

- Reference

[3] 4.4.2.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map



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

- Test Procedure

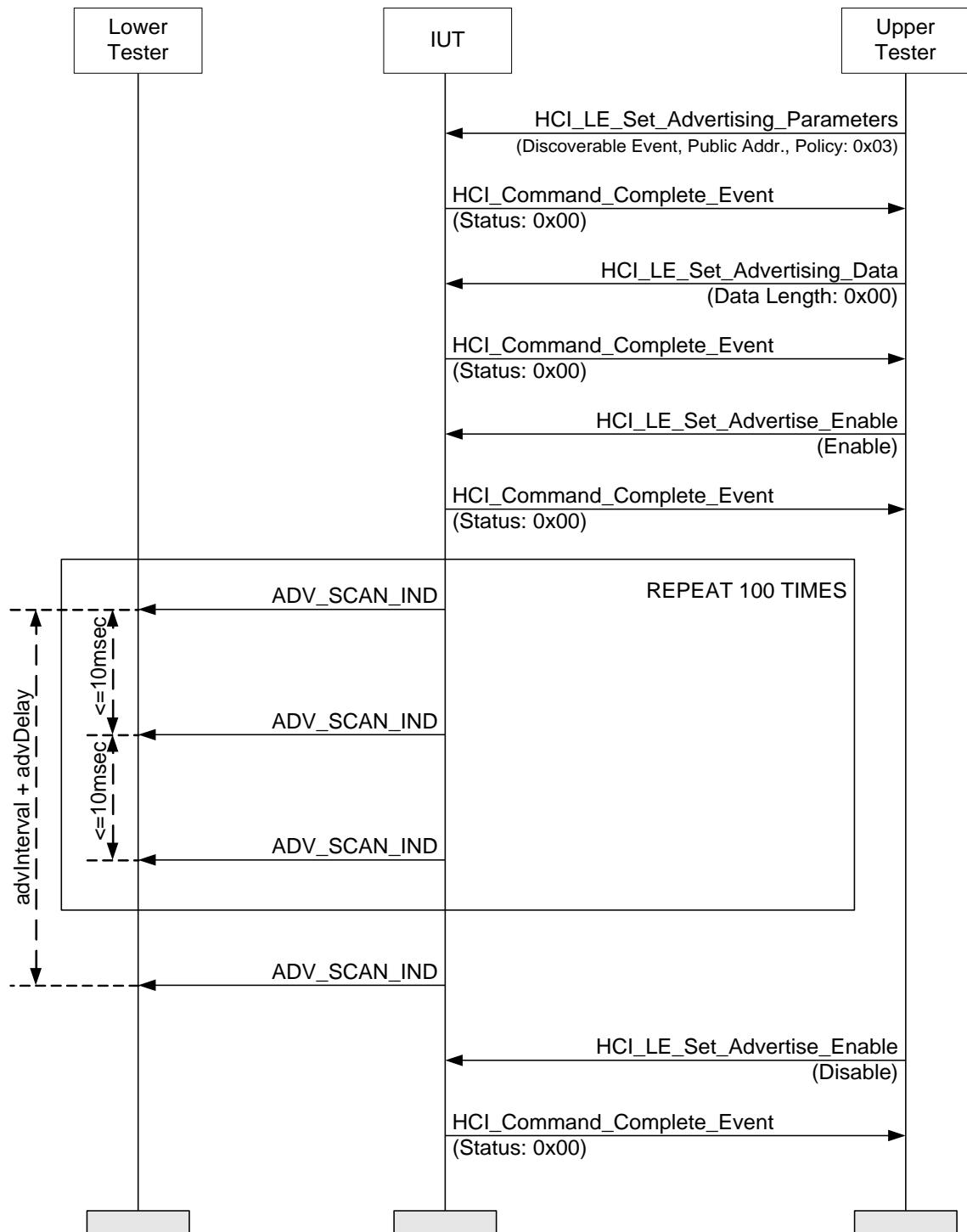


Figure 4.39: LL/DDI/ADV/BV-15-C [Discoverable Advertising Events]



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

Pass Verdict

The test procedure completes using the minimum advertising interval supported.

The test procedure completes using the selected advertising channel.

The timing range detected for advertising events is from (TSPX_adv_interval_min) ms to (TSPX_adv_interval_min + 10) ms.

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

- Test Purpose

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

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

- Reference

[3] 4.4.2.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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



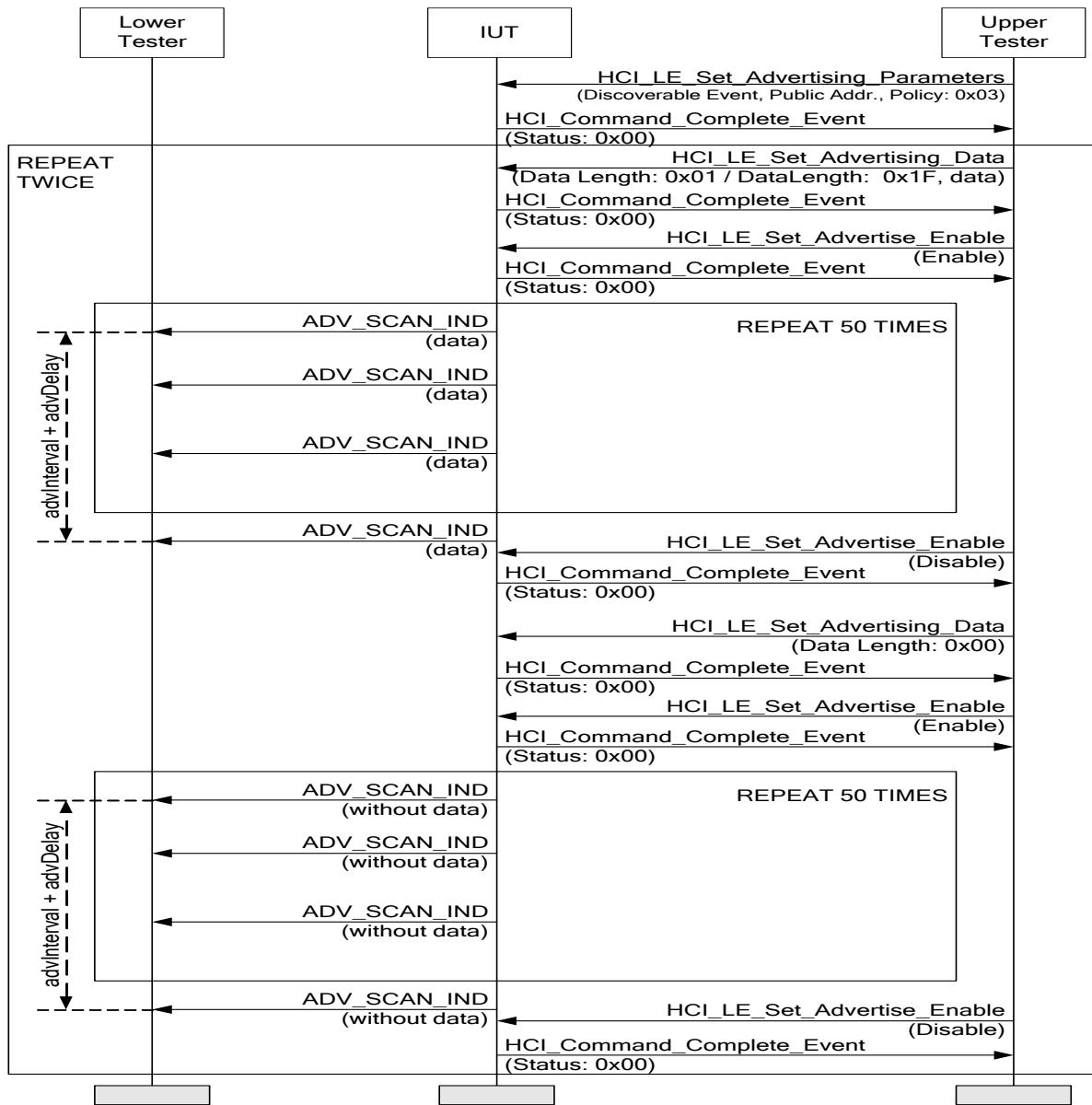


Figure 4.40: LL/DDI/ADV/BV-16-C [Advertising Data: Discoverable]

Configure Lower Tester to monitor advertising packets from the IUT.

1. Upper Tester enables discoverable undirected advertising in the IUT using a selected advertising channel and a selected advertising interval between the minimum and maximum advertising intervals.
2. Upper Tester sends an **HCI LE Set Advertising Data** command to the IUT and expects an **HCI Command Complete** in response. The data element used in the command is a number indicating the length of the data. The data length is 1 byte.
3. Upper Tester sends an **HCI LE Set Advertise Enable** command to the IUT to enable advertising and expects an **HCI Command Complete** event in response.
4. Lower Tester expects the IUT to send **ADV_SCAN_IND** packets including the data submitted in Step 3 starting an event on the selected advertising channel.
5. Expect the following event to start after advertising interval time calculating from the start of the first packet.
6. Repeat Steps 5-6 until a number of advertising intervals (50) have been detected.



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

Pass Verdict

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

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

- Test Purpose

Tests that an advertiser IUT, advertising with the discoverable undirected event type, responds to a scan request and continues advertising after the response.

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

- Reference

[3] 4.4.2.5, 4.4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

Execute the test procedure using the minimum advertising interval supported.



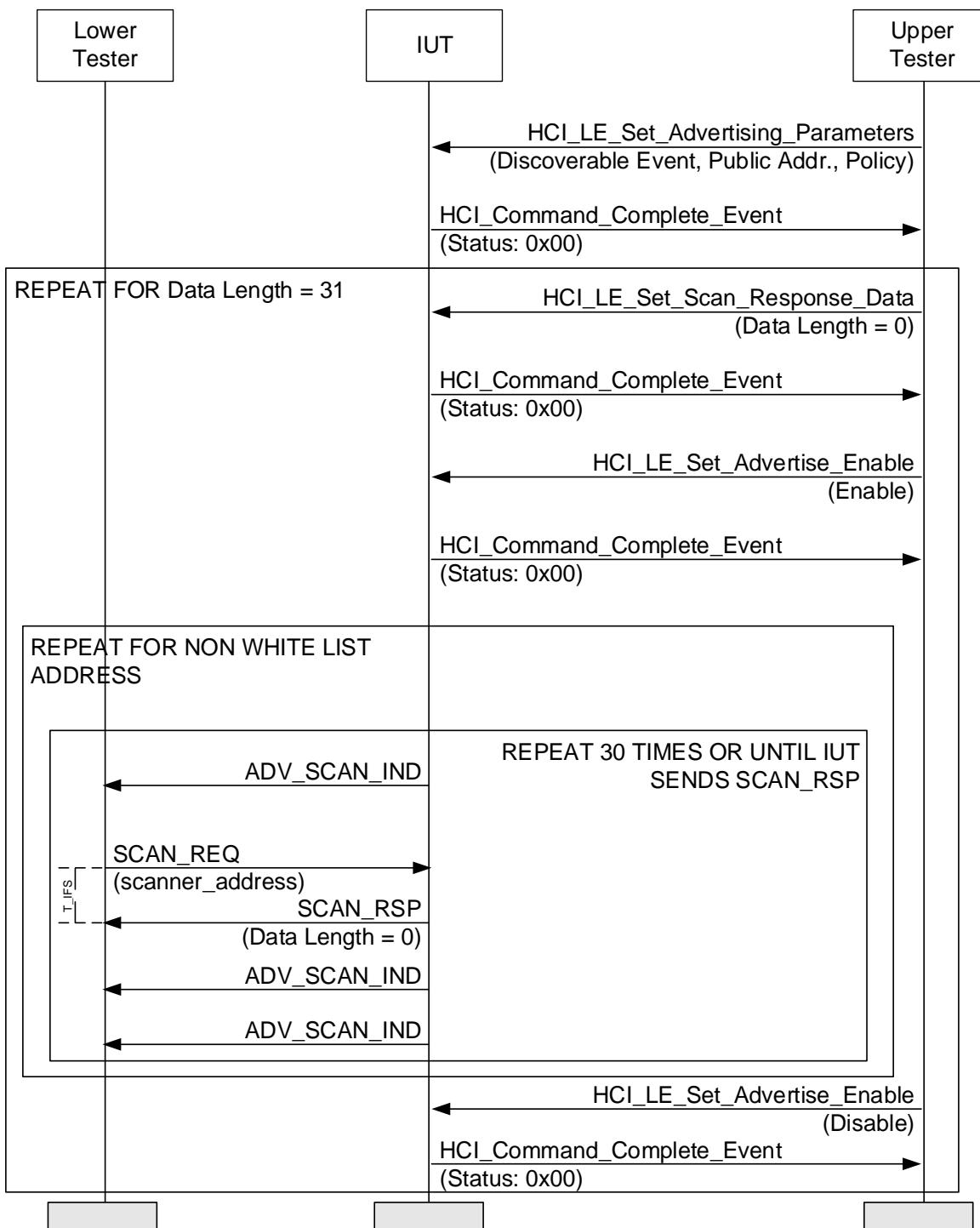


Figure 4.41: LL/DDI/ADV/BV-17-C [Scan Request: Discoverable]

1. Upper Tester enables discoverable undirected advertising in the IUT using all supported advertising channels and minimum advertising intervals.
2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a common device address as parameter.
3. Configure Scan Response Data in the IUT using device name length of 0 as response data.



4. Lower Tester sends a SCAN_REQ packet on the selected advertising channel after receiving an ADV_SCAN_IND packet from IUT on the advertising channel configured in Step 3. The SCAN_REQ is sent T_IFS after the end of an ADV_SCAN_IND packet.
 5. Lower Tester expects a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
 6. Repeat steps 4-5 30 times.
 7. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 8. Repeat steps 4-6.
 9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a device address with identical company assigned identifier to address of the IUT but a different company identifier as parameter.
 10. Configure Scan Response Data in the IUT using device name length of 0 as response data.
 11. Repeat steps 4-6.
 12. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 13. Repeat steps 4-6.
 14. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT. The Lower Tester will send an SCAN_REQ packet on a selected supported advertising channel (defined as an IXIT) and using a device address with address mismatching the IUT address in both the company assigned identifier and the company identifier as parameter.
 15. Configure Scan Response Data in the IUT using device name length of 0 as response data.
 16. Repeat steps 4-6.
 17. Configure Scan Response Data in the IUT using device name length of 31 as response data.
 18. Repeat steps 4-6.
- Expected Outcome

Pass Verdict

The test procedure completes using the minimum advertising interval supported,

The test procedure completes with the IUT responding on each advertising channel,

The IUT responds in each case of different scanner address used,

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

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

- Test Purpose

Tests that an advertiser IUT advertising with the discoverable undirected event type filters scanners according to the white list and filtering policy set.

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

- Reference

[3] 4.3.2, 4.4.2.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map.

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



name) AND Specific White Listed (one white listed device address, one public type address, policy for advertiser, black list all unknown devices)

- Test Procedure

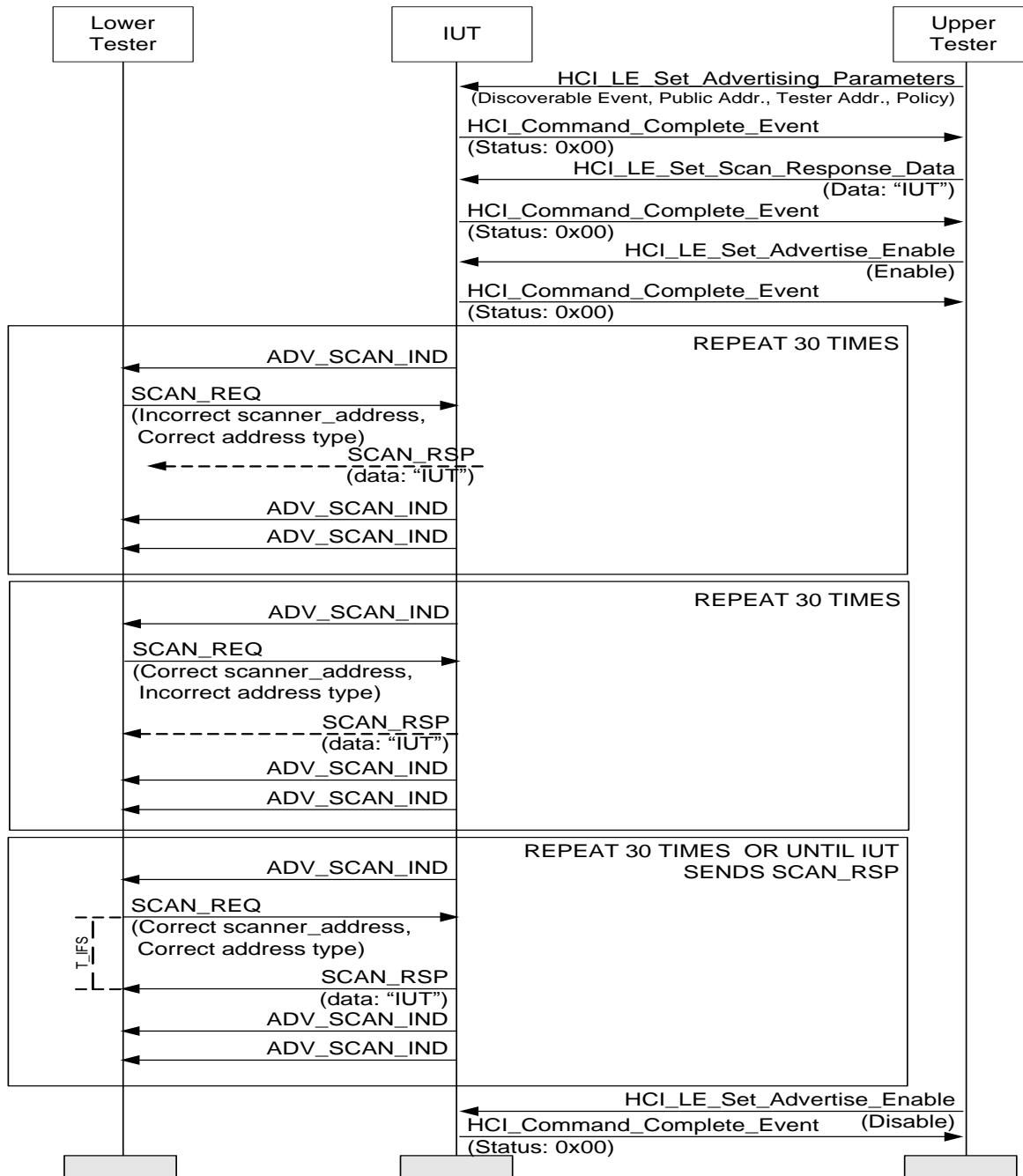


Figure 4.42: LL/DDI/ADV/BV-18-C [Device Filtering: Discoverable]

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



- with an address differing in the company assigned identifier (an address black listed in the policy applied) and correct address type.
4. Lower Tester expects an ADV_SCAN_IND packet from the IUT and responds with an SCAN_REQ packet with the selected address on the selected advertising channel T_IFS after the end of an advertising packet.
 5. Lower Tester expects no response from the IUT.
 6. Repeat steps 4-5 30 times.
 7. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and an incorrect address type.
 8. Repeat steps 4-6 30 times.
 9. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT and send a SCAN_REQ packet on the selected supported advertising channel (defined as an IXIT) with an address white listed in the policy applied and correct address type.
 10. Lower Tester expects an ADV_SCAN_IND packet from the IUT and responds with a SCAN_REQ packet with an address white listed in the policy applied using correct address type, on the selected advertising channel T_IFS after the end of an advertising packet.
 11. Lower Tester expects a SCAN_RSP packet from the IUT addressed to the Lower Tester T_IFS after the end of the request packet.
 12. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.
 13. Upper Tester enables discoverable undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from White List (0x03)'.
 14. Lower Tester address type is set to Random Address Type.
 15. Repeat steps 3-12.
 16. Upper Tester enables discoverable undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from Any (0x01)'.
 17. Lower Tester address type is set to Public Address Type.
 18. Repeat steps 3-12.
 19. Upper Tester enables discoverable undirected advertising in the IUT using public address type, all supported advertising channels, an advertising interval between the minimum and maximum advertising intervals and filtering policy set to 'Allow Scan Request from White List, Allow Connect Request from Any (0x01)'.
 20. Lower Tester address type is set to Random Address Type.
 21. Repeat steps 3-12.
- Expected Outcome

Pass Verdict

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

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

The IUT does not respond to the advertising packets with the white listed addresses,

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



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

- Test Purpose

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

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

- Reference

[3] 3.1, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State:

Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, all advertising channels, Length of device name used, common device name).

- Test Procedure

Execute the test procedure with an advertising interval between the minimum and maximum advertising intervals. For an IUT supporting device filtering, apply the policy of white listing all devices (apply the filtering policy of allowing scan request, connect request from any (0x00)).



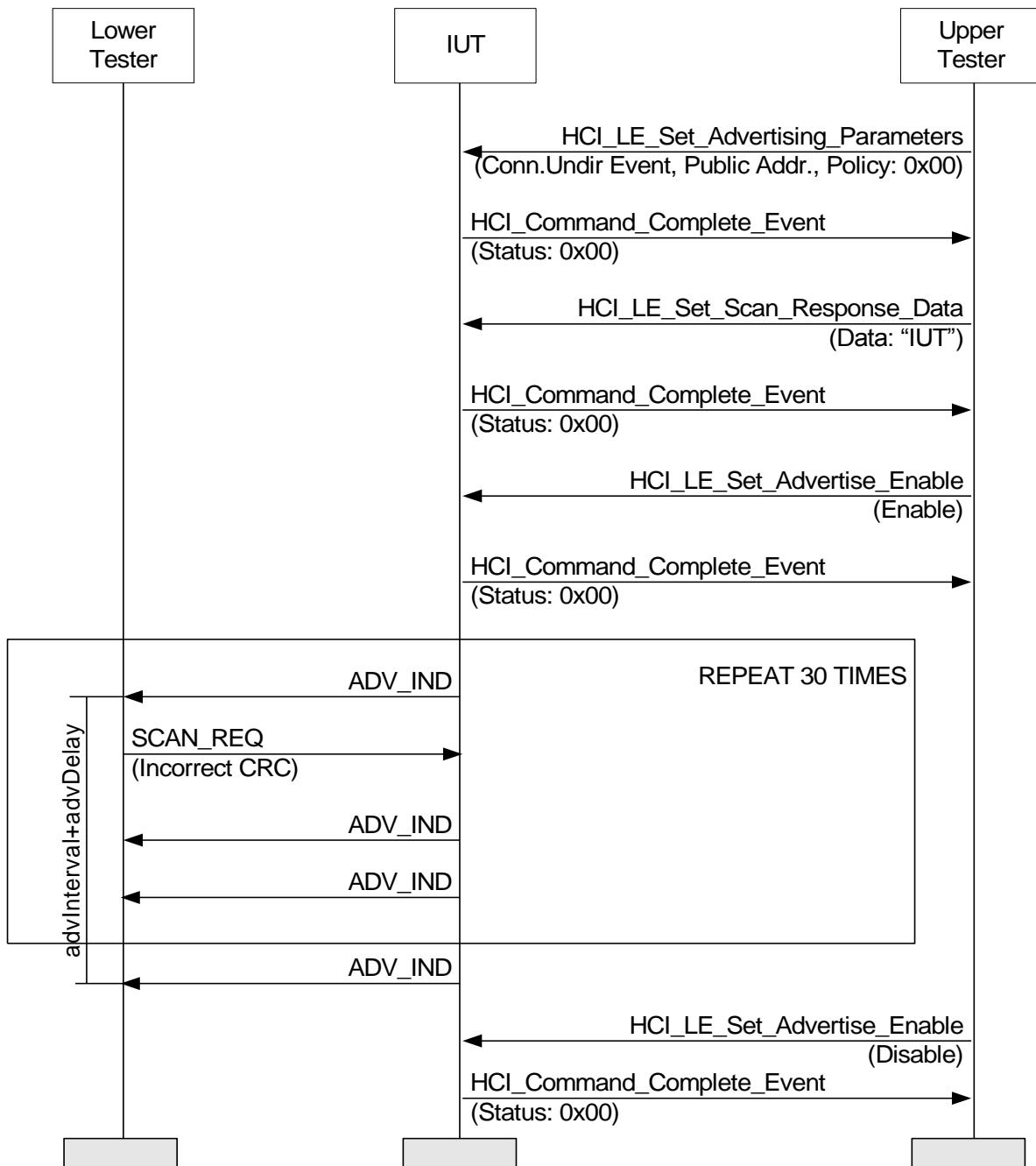


Figure 4.43: LL/DDI/ADV/BI-01-C [Scan Request Invalid CRC]

1. Configure Lower Tester to start an active scanning but sending SCAN_REQ packets with invalid CRC.
 2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
 3. Lower Tester expects an ADV_IND packet from the IUT and responds with an SCAN_REQ packet with an invalid CRC on the selected advertising channel (defined as an IXIT) T_IFS after the end of an advertising packet.
 4. Lower Tester expects the IUT to continue advertising, not responding to the SCAN_REQ packet.
 5. Repeat steps 3-4 30 times.
 6. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.

- Expected Outcome

Pass Verdict

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

4.2.2.17 LL/DDI/ADV/BI-02-C [Connection Request Invalid CRC]

- Test Purpose

Tests that an advertiser IUT ignores connection requests with an invalid CRC.

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

- Reference

[3] 3.1, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

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



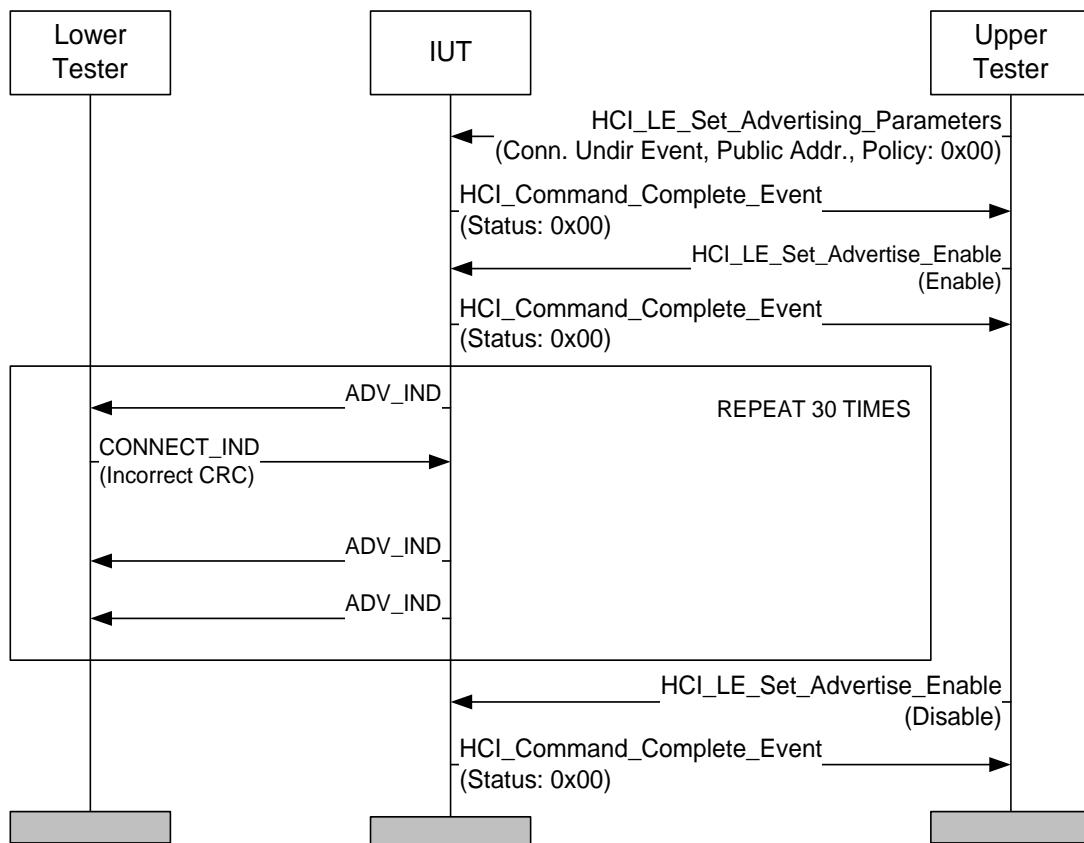


Figure 4.44: LL/DDI/ADV/BV-02-C [Connection Request Invalid CRC]

- Configure Lower Tester to initiate a connection but sending `CONNECT_IND` packets with invalid CRC.
 - Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
 - Lower Tester expects an `ADV_IND` packet from the IUT and responds with a `CONNECT_IND` packet with an invalid CRC on the selected advertising channel T_IFS after the end of an advertising packet.
 - Lower Tester expects the IUT to continue advertising.
 - Repeat steps 3-4 30 times.
 - Upper Tester sends an `HCI_LE_Set_Advertise_Enable` command to disable advertising and expects an `HCI_Command_Complete_Event` in response.
- Expected Outcome

Pass Verdict

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

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

- Test Purpose

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

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



- Reference

[3] 4.4.2.4

- Initial Condition

Parameters: LL_advertiser_advInterval_Min, LL_advertiser_advInterval_Max,
LL_advertiser_Adv_Channel_Map

State: Low Duty Cycle Directed Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, public initiator address, Lower Tester address, selected advertising channels)



- Test Procedure

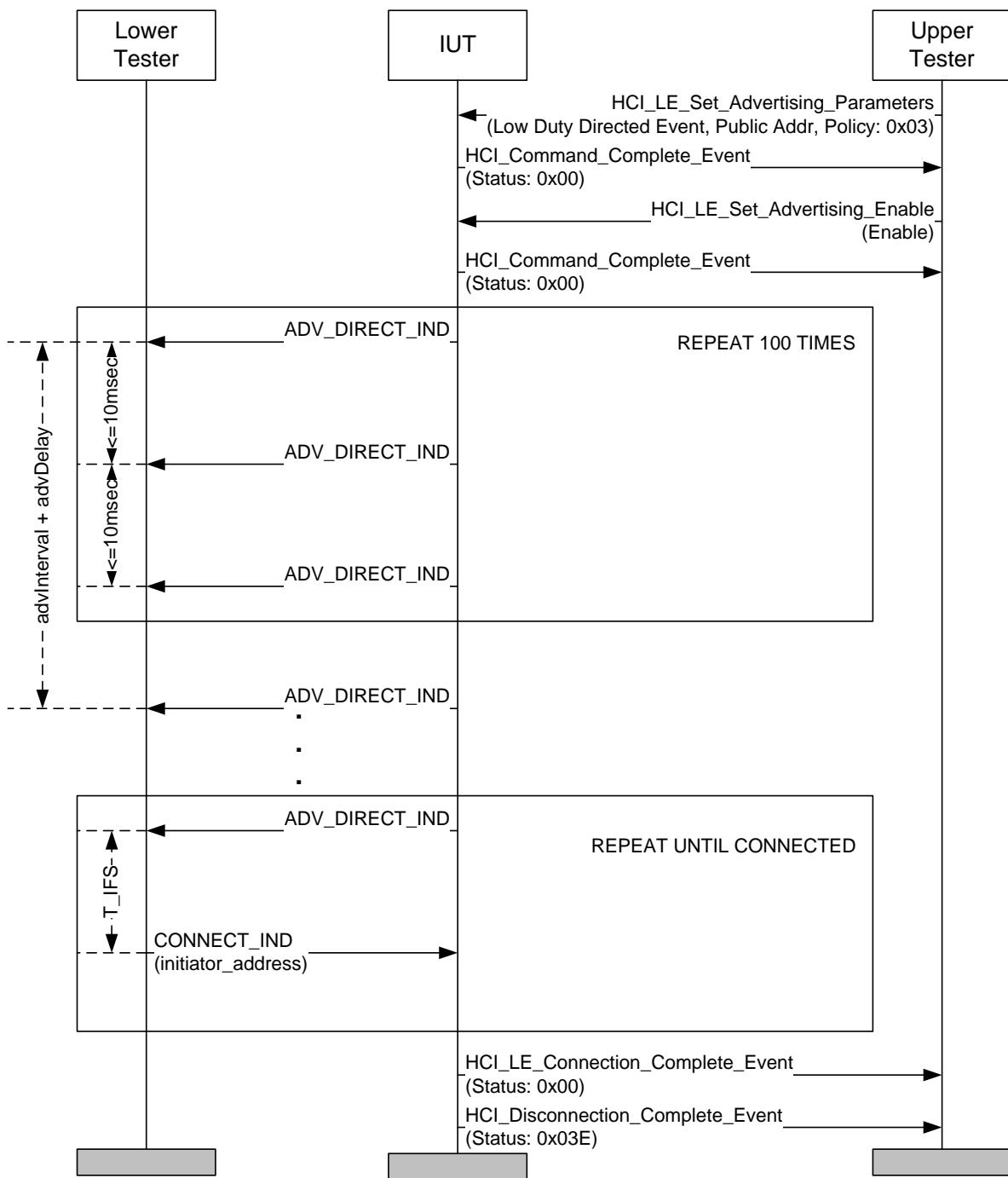


Figure 4.45: LL/DDI/ADV/BV-19-C [Low Duty Cycle Directed Advertising Events]

- Configure Lower Tester to start scanning and monitor advertising packets from the IUT.
- Upper Tester enables low duty cycle directed advertising in the IUT using a selected advertising channel and minimum advertising interval.
- Lower Tester expects the IUT to send ADV_DIRECT_IND packets starting an event on the selected advertising channel.
- Expect the next event to start after the advertising interval time calculated from the start of the first packet.
- Repeat steps 3-4 until the number of advertising intervals (100) have been detected.



6. Configure the Lower Tester to initiate a connection.
7. Lower Tester expects an ADV_DIRECT_IND packet from the IUT on the selected advertising channel (defined as an IXIT), then responds with a CONNECT_IND packet T_IFS after the end of the advertising packet and does not send any data packets to the IUT.
8. Lower Tester expects no ADV_DIRECT_IND packets from the IUT after the advertising interval.
9. Repeat steps 7-8 until the IUT stops advertising.
10. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 7.
11. Upper Tester expects an HCI_Disconnection_Complete event from the IUT once the Establishment Timeout has expired.

- Expected Outcome

Pass Verdict

The test procedure completes using the minimum advertising interval supported.

The timing range detected for advertising events is from 20.0 ms to 30.0 ms (calculated for the minimum advertising interval).

The test procedure completes with the IUT stopping advertising after receiving a connection request on the selected advertising channel.

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

- Notes

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

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

- Test Purpose

Test that an advertiser IUT sends advertising packets of an event with correct contents on all applicable advertising channels using the LE 1M PHY, even when the host has indicated that it prefers the LE 2M PHY.

The Lower Tester observes the packet and event timing and packet contents on the advertising channels used and confirms they can be received using the LE 1M PHY.

- Reference

[10] 2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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



- Test Procedure

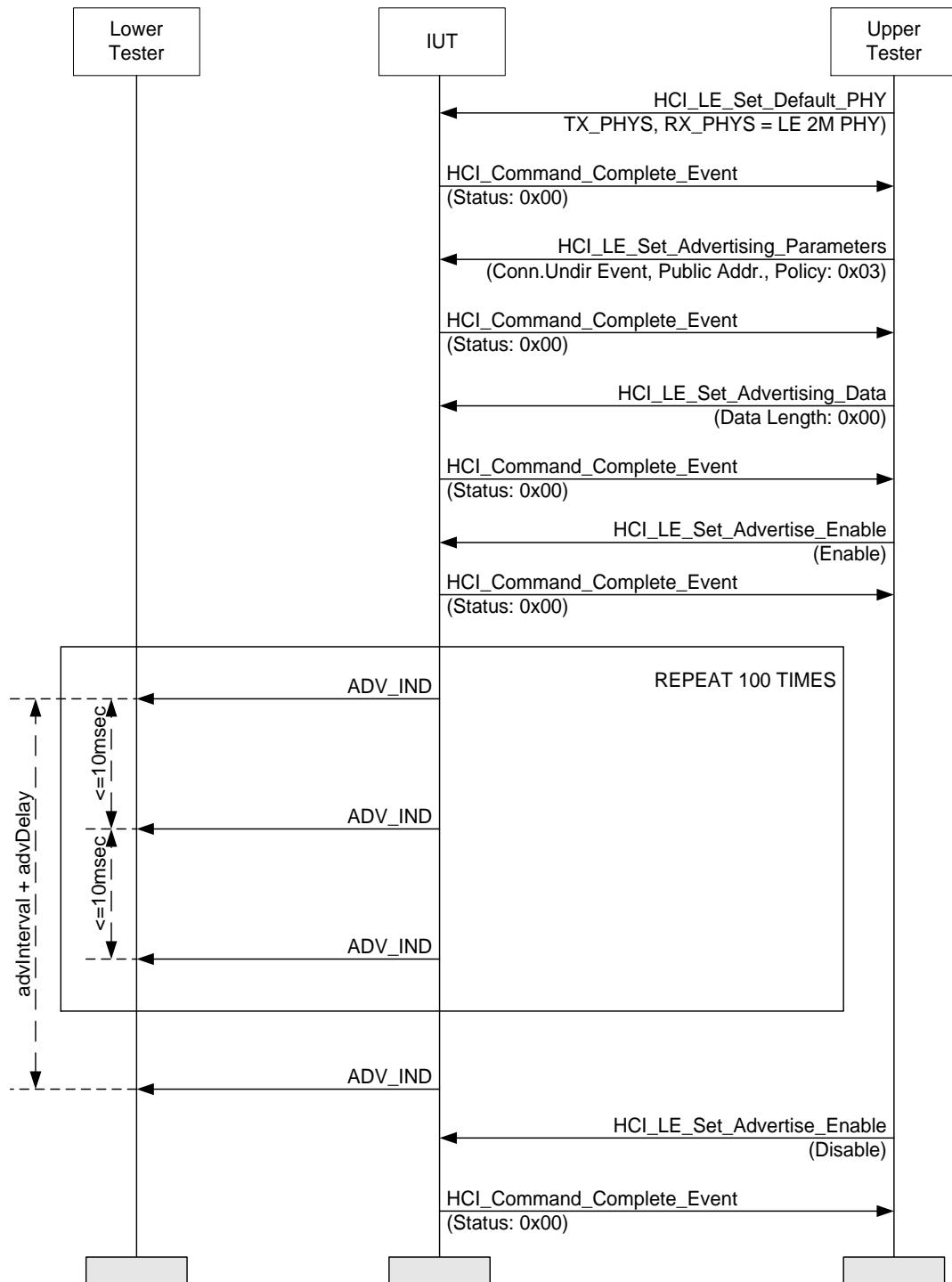


Figure 4.46: LL/DDI/ADV/BV-20-C [Advertising Always Using the LE 1M PHY]

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



2. Upper Tester sends a LE_Set_Default_PHY command to the IUT, with the ALL_PHYS field set to zero, and the TX_PHYS and RX_PHYS fields both set to prefer the LE 2M PHY.
 3. Upper Tester enables undirected advertising in the IUT using all supported advertising channels and minimum advertising interval
 4. Lower Tester expects the IUT to send ADV_IND packets starting an event on an applicable advertising channel using the LE 1M PHY.
 5. Repeat Step 4 until at least 90 advertising packets have been detected, i.e., at least 30 packets on each channel.
 6. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
- Expected Outcome

Pass Verdict

All advertising events are transmitted using the LE 1M PHY and are received properly by the Lower Tester using the LE 1M PHY.

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

- Test Purpose

Tests that an advertiser IUT sends advertising packets of a non-connectable event type with data on a selected primary advertising channel using legacy PDU types and extended advertising HCI commands.

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

- Reference

[\[10\]](#) 4.4.2.6

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

Execute the test procedure using non-connectable advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using a selected primary advertising channel, with data lengths of 1, 0, and 31.



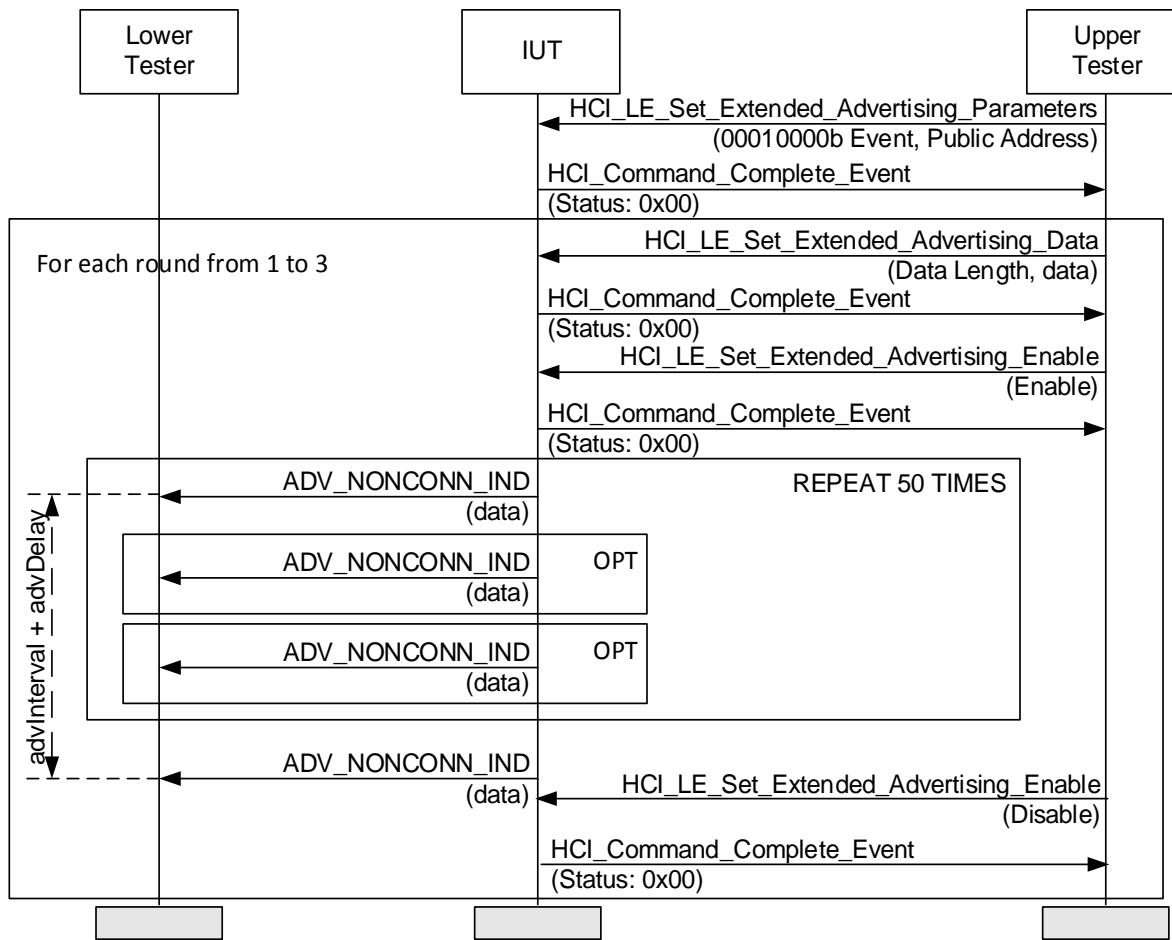


Figure 4.47: LL/DDI/ADV/BV-21-C [Extended Advertising, Legacy PDUs, Non-Connectable]

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

For each round from 1 to 3 based on [Table 4.2](#):

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



Round	HCI_LE_Set_Extended_Advertising_Data (Step 3)	
	Data Length	Data Element
1	1	0x01
2	0	No data
3	31	0xF8, [0x00]...

Table 4.2: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

The test procedure executes with the IUT advertising using non-connectable event type. The ADV_NONCONN_IND PDU is utilized.

The IUT transmits data as submitted in the HCI commands.

4.2.2.21 LL/DDI/ADV/BV-22-C [Extended Advertising, Legacy PDUs, Undirected]

- Test Purpose

Tests that an advertiser IUT sends advertising packets of an undirected type of event with data on all advertising channels using legacy PDU types and extended advertising HCI commands.

The Upper Tester submits data to the IUT, and the Lower Tester observes the IUT including data in the advertising packets on the advertising channels used. The Lower Tester confirms the IUT sets ChSel = 1 in the legacy PDU.

- Reference

[10] 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

Execute the test procedure using undirected advertising events with a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels, with data lengths of 1, 0, and 31.



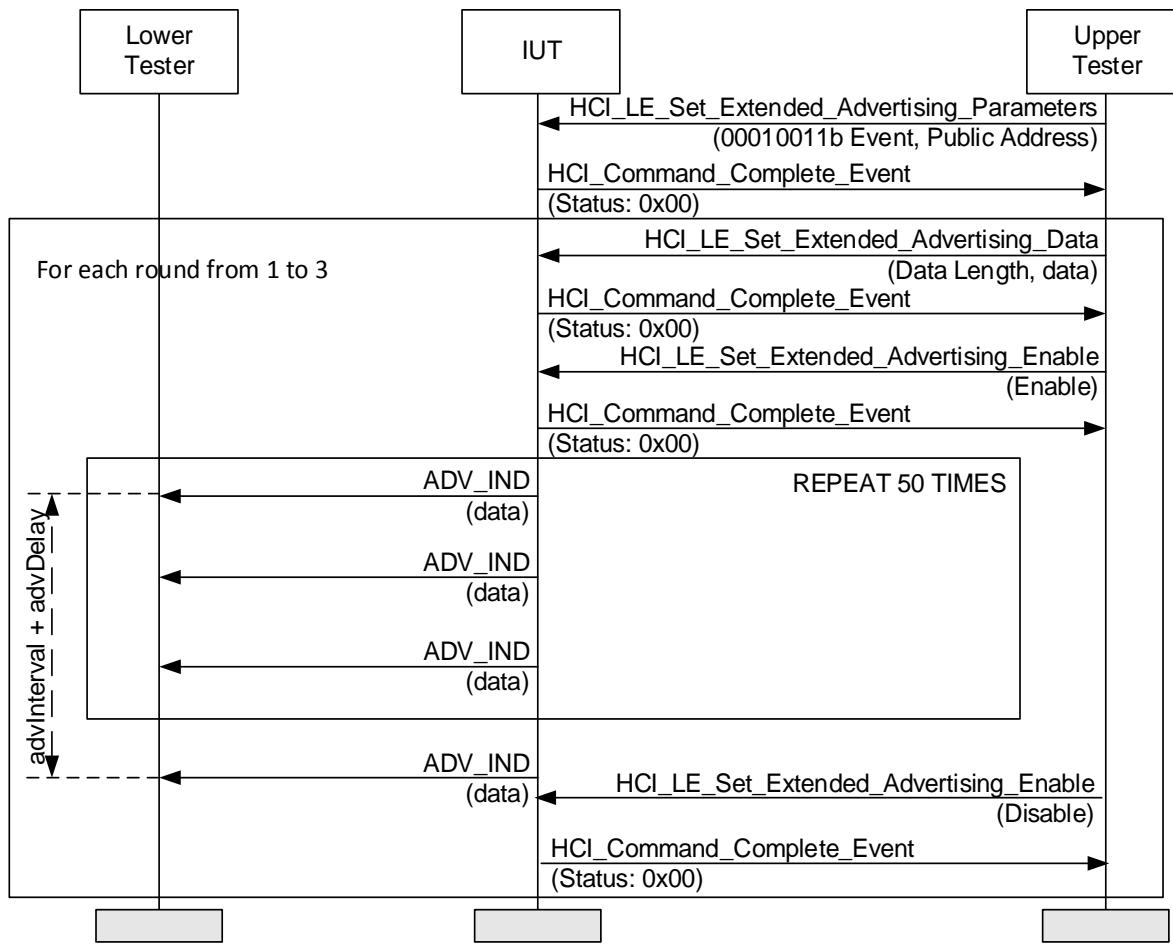


Figure 4.48: LL/DDI/ADV/BV-22-C [Extended Advertising, Legacy PDUs, Undirected]

1. Configure Lower Tester to monitor advertising packets from the IUT.
2. The Upper Tester sends an **HCI_LE_Set_Extended_Advertising_Parameters** command to the IUT using all supported advertising channels and minimum advertising interval. The **Advertising_Event_Properties** parameter shall be set to 00010011b (ADV_IND legacy PDU).

For each round from 1 to 3 based on [Table 4.3](#):

3. Upper Tester sends an **HCI_LE_Set_Extended_Advertising_Data** command to the IUT with values according to [Table 4.3](#) and expects an **HCI_Command_Complete** in response.
4. Upper Tester sends an **HCI_Command_Complete** event to the IUT to enable advertising with Duration[0] set to 0x0000 (continue advertising until disabled), and expects an **HCI_Command_Complete** event in response.
5. Lower Tester scans on a single primary advertising channel as indicated in [Table 4.3](#) and expects the IUT to send ADV_IND packets, with ChSel set to 1, including the data submitted in step 3 starting an event on the applicable primary advertising channel.
6. Expect the following event to start after advertising interval time calculating from the start of the first packet.
7. Repeat Steps 5–6 until a number of advertising intervals (50) have been detected
8. Upper Tester sends an **HCI_Command_Complete** event to the IUT to disable advertising function and expects an **HCI_Command_Complete** event in response.
9. Repeat Steps 3–8 for each Round shown in [Table 4.3](#).

Round	HCI_LE_Set_Extended_Advertising_Data (Step 3)		
	Data Length	Data Element	Primary Advertising Channel scanned by the Lower Tester
1	1	0x01	37
2	0	No data	38
3	31	0xF8, [0x00]...	39

Table 4.3: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

The test procedure executes with the IUT advertising using Connectable Undirected event type. The ADV_IND PDU is utilized, with ChSel set to 1.

The IUT transmits data as submitted in the HCI commands.

4.2.2.22 Extended Advertising, Non-Connectable

- Test Purpose

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

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

- Reference

[10] 4.4.2.6, 4.4.2.10

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

The following test procedure applies to the test cases listed in [Table 4.5](#), the only change is the variation of the PHYs used.

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



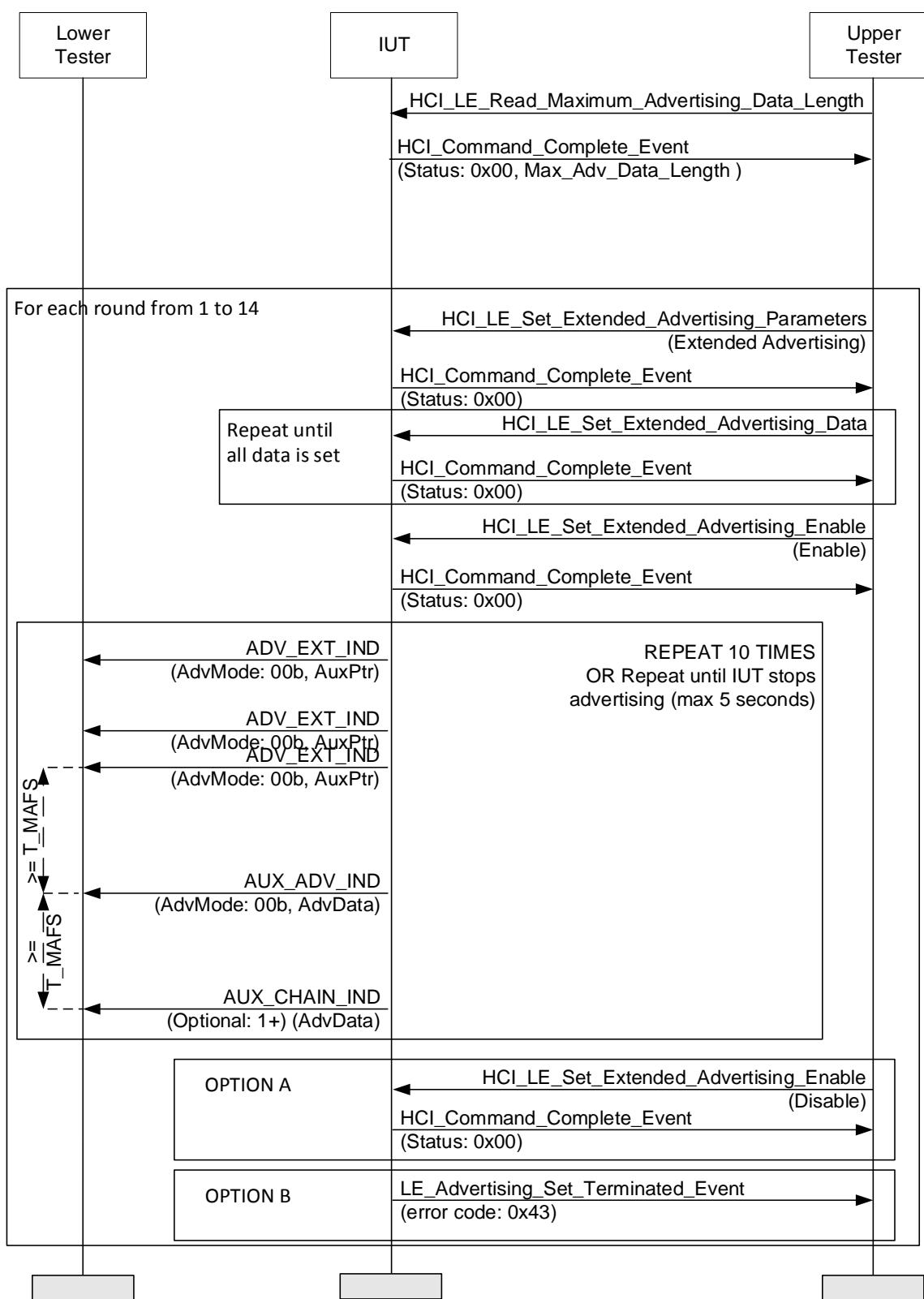


Figure 4.49: Extended Advertising, Secondary Channel, Non-Connectable

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



For each round based on [Table 4.4](#):

2. If the Data Length listed in [Table 4.4](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to step 3, otherwise skip to step 14.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to the value specified in [Table 4.4](#) for this round. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY and Secondary_Advertising_PHY shall be set to the values specified in [Table 4.5](#). If the Advertising_Event_Properties value for this Round specifies directed advertising, the Peer_Address_Type shall be set to 0x00 (Public Device Address), and the Peer_Address shall be set to the Lower Tester's address.
4. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT with values according to [Table 4.4](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter shall be set to the value specified in [Table 4.4](#) for this round. The Max_Extended_Advertising_Events[0] parameter shall be set to the value specified in [Table 4.4](#) for this round
6. The Lower Tester expects an ADV_EXT_IND packet from the IUT with AdvMode set to 00b. The ADV_EXT_IND PDU shall not include the SuplInfo, SyncInfo, TxPower, ACAD, or AdvData fields. If advertising data was set in step 4, the ADV_EXT_IND PDU shall include the AuxPtr field; otherwise, the ADV_EXT_IND PDU may include the AuxPtr field. If the AuxPtr field is included, the ADV_EXT_IND PDU shall also include the ADI field with the SID set to the value used in step 3; otherwise that field shall not be included.
7. If the AuxPtr is absent, skip to step 10.
8. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b. The AUX_ADV_IND PDU shall not include the SuplInfo, SyncInfo, or TxPower fields. The AUX_ADV_IND PDU shall include the ADI field matching the ADI field from step 6. If the AUX_ADV_IND PDU does not contain all the data submitted in step 4 (if any), it shall include an AuxPtr field.
9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall include the ADI field matching the ADI field from step 6 and the AdvData field containing additional data submitted in step 4. The AUX_CHAIN_IND PDU shall not include the AdvA, TargetA, SuplInfo, TxPower, or SyncInfo fields. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.
10. If the Max_Extended_Advertising_Events was set to a value different than 0, repeat steps 6–9 until the IUT stops advertising. Afterwards, the Lower Tester confirms that the IUT did not send more than Max_Extended_Advertising_Events advertising events. Upper Tester shall receive LE Advertising Set Terminated event with ErrorCode 0x43. Skip to step 13.
11. Otherwise if Duration was set to a value different than 0, repeat steps 6–9 until the amount of time specified for Duration has elapsed. Afterwards, the Lower Tester confirms that the IUT does not start any additional advertising events. Upper Tester shall receive LE Advertising Set Terminated event with ErrorCode 0x3C. Skip to step 13.
12. Otherwise, repeat steps 6–9 until a number of advertising intervals (10) have been detected.



13. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. Repeat steps 2–13 for each Round shown in [Table 4.4](#).

Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3)	HCI_LE_Set_Extended_Advertising_Data (Step 4)		HCI_LE_Set_Extended_Advertising_Enable (Step 5)	
	Advertising_Event_Properties	Data Length	Fragment_Preference	Duration	Max_Extended_Advertising_Event_S
1	0x0000	0	0x00	0x0000	0x00
2	0x0000	31	0x00	0x0000	0x00
3	0x0000	474	0x00	0x0000	0x00
4	0x0000	711	0x00	0x0000	0x00
5	0x0000	948	0x00	0x0000	0x00
6	0x0000	Maximum_Advertising_Data_Length	0x00	0x0000	0x00
7	0x0000	Maximum_Advertising_Data_Length	0x01	0x0000	0x00
8	0x0004	0	0x00	0x0000	0x00
9	0x0004	251	0x00	0x0000	0x00
10	0x0004	Maximum_Advertising_Data_Length	0x00	0x0000	0x00
11	0x0000	0	0x00	0x01F4 (5 s)	0x00
12	0x0004	0	0x00	0x01F4 (5 s)	0x00
13	0x0000	0	0x00	0x0000	0x32
14	0x0004	0	0x00	0x0000	0x32

Table 4.4: Payload contents for each case variation

- Expected Outcome

The following test procedure applies to the test cases listed in Table 4.5, the only change is the variation of the PHYs used.

Pass Verdict

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

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

- The IUT sends an ADV_EXT_IND PDU on the primary advertising channel.
- If advertising data was specified for the round, the ADV_EXT_IND PDU includes the AuxPtr field referring to an AUX_ADV_IND PDU on the secondary channel.



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

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

Table 4.5: Extended Advertising, Non-Connectable test cases

- Notes
 - The minimum actual advertising duration required is the duration specified plus negative drift plus negative jitter: $(10 \text{ ms} * \text{Duration}) - (10 \text{ ms} * \text{Duration} * 500 \text{ ppm}) - 0.016 \text{ ms}$.



- The maximum actual duration is $(10 \text{ ms} * \text{Duration} + 10.000 \text{ ms}) * (1 + 500 \text{ ppm}) + 0.016 \text{ ms}$.
- The notes in [LL/DDI/ADV/BV-01-C \[Non-Connectable Advertising Events\]](#) describe the reasoning of the timing measurements and test result criteria.

4.2.2.23 [Extended Advertising, Scannable]

- Test Purpose

Tests that an advertiser IUT sends scannable ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel with the correct payload fields, timing, and channel sequence for the maximum time allowed. Tests that an advertiser IUT responds to a scan request on the secondary channel and continues advertising after the response. Scan response data chaining is tested. Undirected and Directed events are tested.

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

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

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



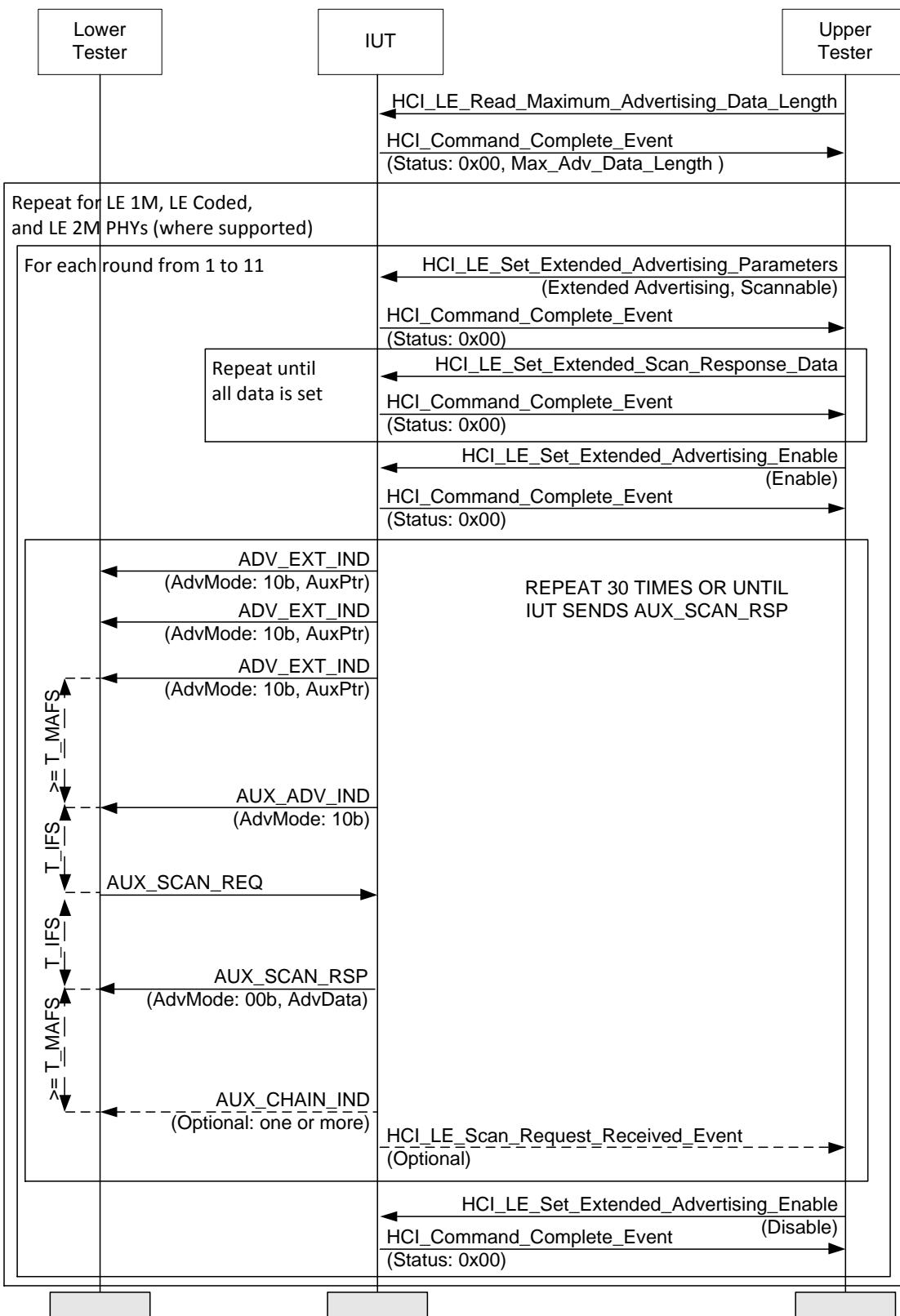


Figure 4.50: LL/DDI/ADV/BV-25-C [Extended Advertising, Secondary Channel, Scannable]

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



For each round from 1 to 11 based on [Table 4.6](#):

2. If the Data Length listed in [Table 4.6](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 13.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties and Scan_Response_Notification_Enable parameters shall be set to the value specified in [Table 4.6](#) for this round. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M). If the Advertising_Event_Properties value for this Round specifies directed advertising, the Peer_Address_Type shall be set to 0x00 (Public Device Address), and the Peer_Address shall be set to the Lower Tester's address.
4. The Upper Tester sends one or more HCI_LE_Set_Extended_Scan_Response_Data commands to the IUT with values according to [Table 4.6](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
6. The Lower Tester expects an ADV_EXT_IND packet from the IUT with AdvMode set to 10b with the AuxPtr Extended Header field present. The ADV_EXT_IND PDU shall include the ADI field with the SID set to the value used in step 3. The ADV_EXT_IND PDU shall not include the SuplInfo, SyncInfo, TxPower, ACAD, or TxPower fields.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 10b. The AUX_ADV_IND PDU shall include the ADI field matching the ADI field from step 6. The AUX_ADV_IND PDU shall not include the SuplInfo, AuxPtr, SyncInfo, TxPower, or AdvData fields.
8. The Lower Tester responds with an AUX_SCAN_REQ PDU TIFS after the end of the AUX_ADV_IND PDU on the secondary advertising channel with ScanA set to the Lower Tester's address and AdvA set as shown in [Table 4.6](#).
9. If the AUX_SCAN_REQ PDU has an AdvA not equal to the IUT's address, the Lower Tester expects no AUX_SCAN_RSP packet from the IUT. Skip to Step 12.
10. The Lower Tester expects an AUX_SCAN_RSP packet from the IUT TIFS after the end of the AUX_SCAN_REQ PDU with the fields in [Table 4.7](#). If the AUX_SCAN_RSP PDU does not contain all the data submitted in Step 4 (if any), it shall include an AuxPtr field.
11. If the AUX_SCAN_RSP PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall include the AdvData field containing additional data submitted in Step 4. The AUX_CHAIN_IND PDU shall not include the AdvA, TargetA, SuplInfo, TxPower, or SyncInfo fields. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.
12. If the Scan_Response_Notification_Parameter value specified for step 3 in this round was 0x01 (notifications enabled), the Upper Tester expects an HCI_LE_Scan_Request_Received event from the IUT with the advertising handle used in steps 3–5 and the Lower Tester's address from step 8.
13. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. Repeat steps 2–13 for each Round shown in [Table 4.6](#).



15. If the IUT supports the LE Coded PHY, repeat steps 2–14, except that in Step 3 the Primary_Advertising_PHY shall be set to 0x03 (LE Coded) and the Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
16. If the IUT supports the LE 2M PHY, repeat steps 2–14, except that in Step 3 the Primary_Advertising_PHY shall be set to 0x01 (LE 1M) and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).

Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3)		HCI_LE_Set_Extended_Scan_Response_Data (Step 4)		AUX_SCAN_REQ_PDU (Step 8)
	Advertising_Event_Properties	Scan_Response_Notification_Enable	Data Length	Fragment_Preference	AdvA
1	0x0002	0x00	1	0x00	IUT
2	0x0002	0x00	31	0x00	IUT
3	0x0002	0x00	474	0x00	IUT
4	0x0002	0x00	711	0x00	IUT
5	0x0002	0x00	948	0x00	IUT
6	0x0002	0x00	Maximum_Advertising_Data_Length	0x00	IUT
7	0x0002	0x01	Maximum_Advertising_Data_Length	0x01	IUT
8	0x0002	0x00	31	0x00	Not IUT
9	0x0006	0x00	1	0x00	IUT
10	0x0006	0x00	251	0x00	IUT
11	0x0006	0x00	Maximum_Advertising_Data_Length	0x00	IUT
12	0x0006	0x00	31	0x00	Not IUT

Table 4.6: Payload contents for each case variation



- Test Case Configuration

Test Case	Reference	AdvMode	AdvA	Target A	SupplInfo	ADI Field
4.2.2.23.1 LL/DDI/ADV/BV-25-C [Extended Advertising, Scannable – without ADI]	[10] 4.4.2.5.2, 4.4.2.8	00b	Step 10: IUT advertising address from step 3. Pass verdict: IUT advertising address specified in the HCI_LE_Set_Extended_Advertising_Parameters	Not Present	Not Present	Not Present
4.2.2.23.2 LL/DDI/ADV/BV-45-C [Extended Advertising, Scannable – with ADI]	[10] 4.4.2.5.2, 4.4.2.8, [13] 2.3.2.3	00b	Step 10: IUT advertising address from step 3. Pass verdict: IUT advertising address specified in the HCI_LE_Set_Extended_Advertising_Parameters	Not Present	Not Present	ADI Field in AUX_ADV_IND PDU

Table 4.7: Extended Advertising, Scannable Test Cases

- Expected Outcome

Pass Verdict

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

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

- The IUT sends an ADV_EXT_IND PDU on the primary advertising channel with an AuxPtr field referring to an AUX_ADV_IND PDU on the secondary advertising channel.
- The IUT responds to the AUX_SCAN_REQ within the 2 μ s range around T_IFS.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- The ADV_EXT_IND PDU does not include the SupplInfo, SyncInfo, TxPower, ACAD, or TxPower fields.
- The AUX_ADV_IND PDU does not include the SupplInfo, AuxPtr, SyncInfo, TxPower, or AdvData fields.
- The ADV_EXT_IND and AUX_ADV_IND PDUs both contain the ADI field with the SID set to the value specified in the HCI_LE_Set_Extended_Advertising_Parameters command.
- If scan response data was specified for the round, the Lower Tester receives all the data sent in step 4, and no further data, in the AUX_SCAN_RSP PDU and zero or more AUX_CHAIN_IND PDUs. If no data was specified, the AUX_SCAN_RSP PDU and any AUX_CHAIN_IND PDUs shall not contain any data.
- If the scan request notifications were enabled for the round, the Upper Tester receives an HCI_LE_Scan_Request_Received event from the IUT with the advertising handle and the Lower Tester's address.
- If the AUX_SCAN_REQ PDU has an AdvA not equal to the IUT's address then the IUT shall not respond.



- The AUX_SCAN_RSP PDU includes or excludes the fields as specified in [Table 4.7](#).
- If one or more AUX_CHAIN_IND PDUs were sent, none of them include the AdVA, TargetA, SuplInfo, ADI, TxPower, or SyncInfo fields.

4.2.2.24 LL/DDI/ADV/BV-26-C [Extended Advertising, Periodic Advertising]

- **Test Purpose**

Tests that an advertiser IUT sends ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU containing a SyncInfo field that further refers to AUX_SYNC_IND PDUs on the secondary advertising channel with the correct payload fields, timing, and channel sequence.

Advertisements with and without data, along with chaining, are tested. The Lower Tester confirms that Channel Selection Algorithm #2 is utilized for the periodic advertisements.

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

- **Reference**

[\[10\] 4.4.2.13.1](#)

- **Initial Condition**

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- **Test Procedure**

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



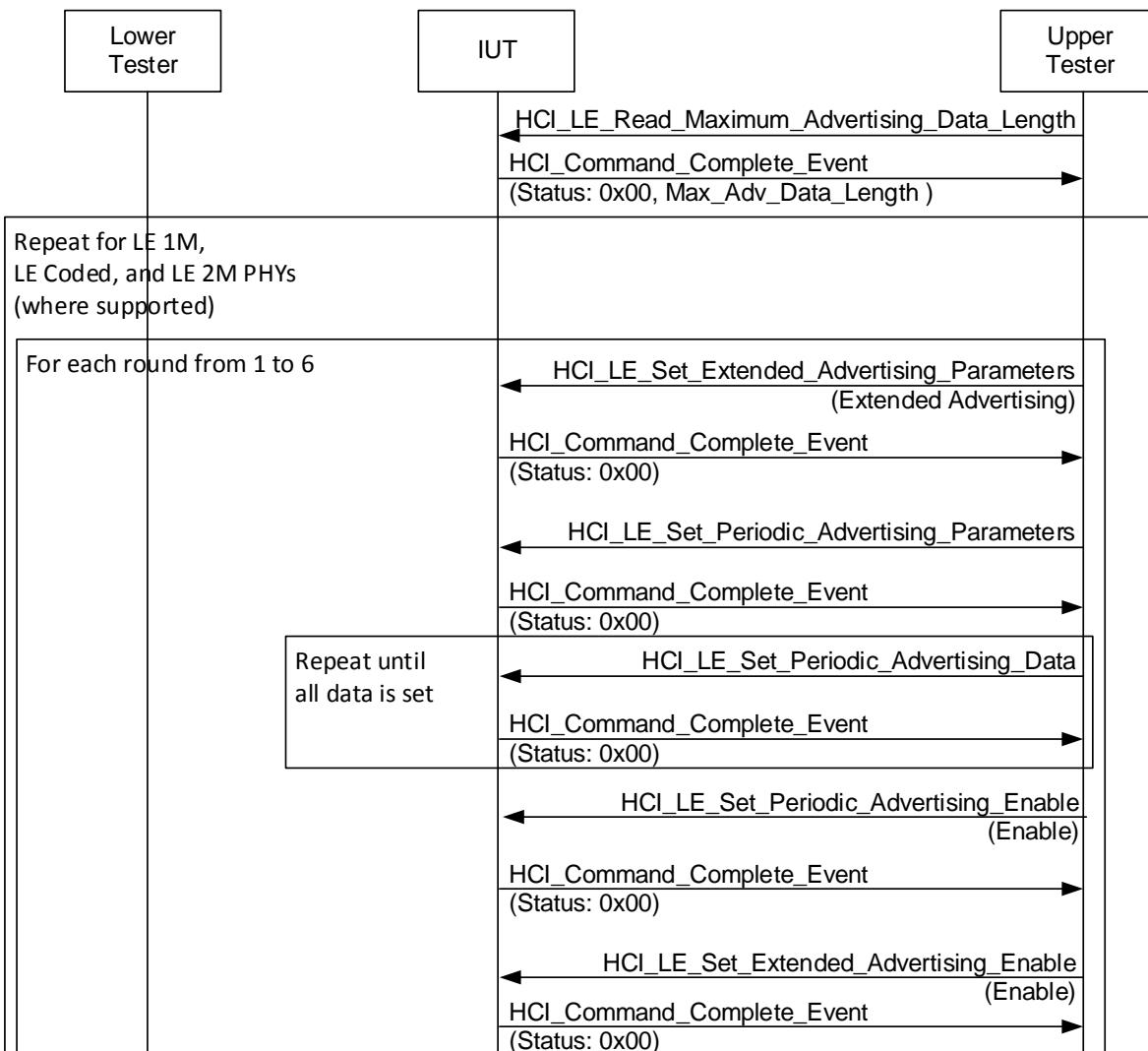


Figure 4.51: LL/DDI/ADV/BV-26-C [Extended Advertising, Periodic Advertising] – Part A



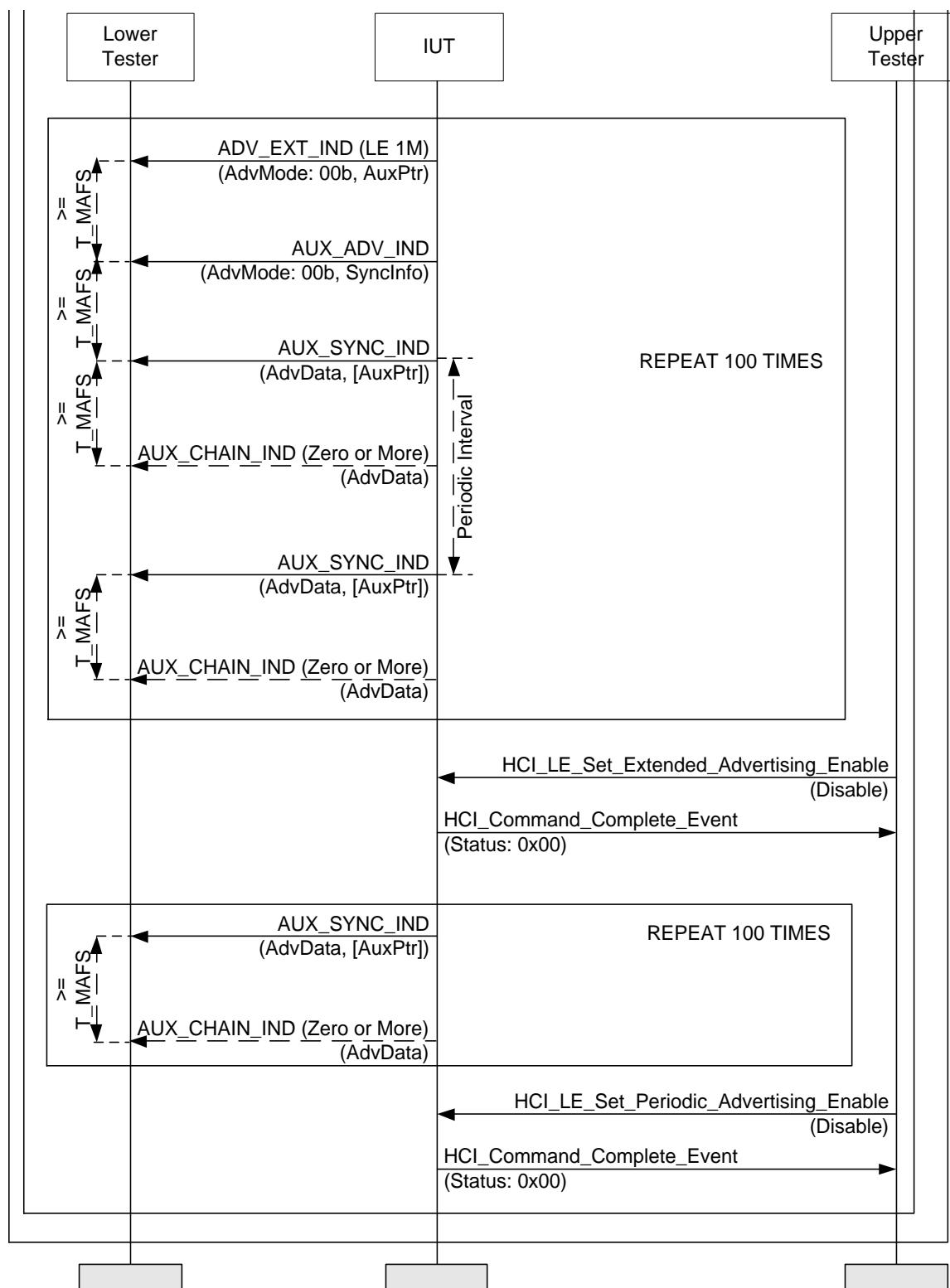


Figure 4.52: LL/DDI/ADV/BV-26-C [Extended Advertising, Periodic Advertising] – Part B

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

For each round from 1 to 6 based on [Table 4.8](#):

2. If the Data Length listed in [Table 4.8](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 16.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.
5. The Upper Tester sends one or more HCI_LE_Set_Periodic_Advertising_Data commands to the IUT with values according to [Table 4.8](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 252 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
6. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (Periodic Advertising).
7. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
8. The Lower Tester expects an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present.
9. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields present.
10. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the secondary advertising channel using the index selected by the LE Channel Selection Algorithm #2 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall have the AdvMode field set to 00b. If the AUX_SYNC_IND PDU does not contain all the data submitted in Step 5 (if any), it shall include an AuxPtr field.
11. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in Step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.
12. Repeat Steps 8-11 100 times.
13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command but maintains synchronization with the IUT's periodic advertising.
14. The Lower Tester confirms that periodic advertising continues when extended advertising is disabled by repeating Steps 10-11 100 times.
15. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
16. Repeat steps 2-15 for each Round shown in [Table 4.8](#).
17. If the IUT supports the LE Coded PHY, repeat steps 2-16, except that in Step 3 the Primary_Advertising_PHY shall be set to 0x03 (LE Coded) and the Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).



18. If the IUT supports the LE 2M PHY, repeat steps 2-16, except that in Step 3 the Primary_Advertising_PHY shall be set to 0x01 (LE 1M) and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).

Round	HCI_LE_Set_Periodic_Advertising_Data (Step 5)	
	Data Length	
1	0	
2	252	
3	474	
4	711	
5	948	
6	Maximum_Advertising_Data_Length	

Table 4.8: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

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

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

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

- Test Purpose

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

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



- Reference
[\[10\]](#) 4.4.2.6, 2.3.4.4

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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



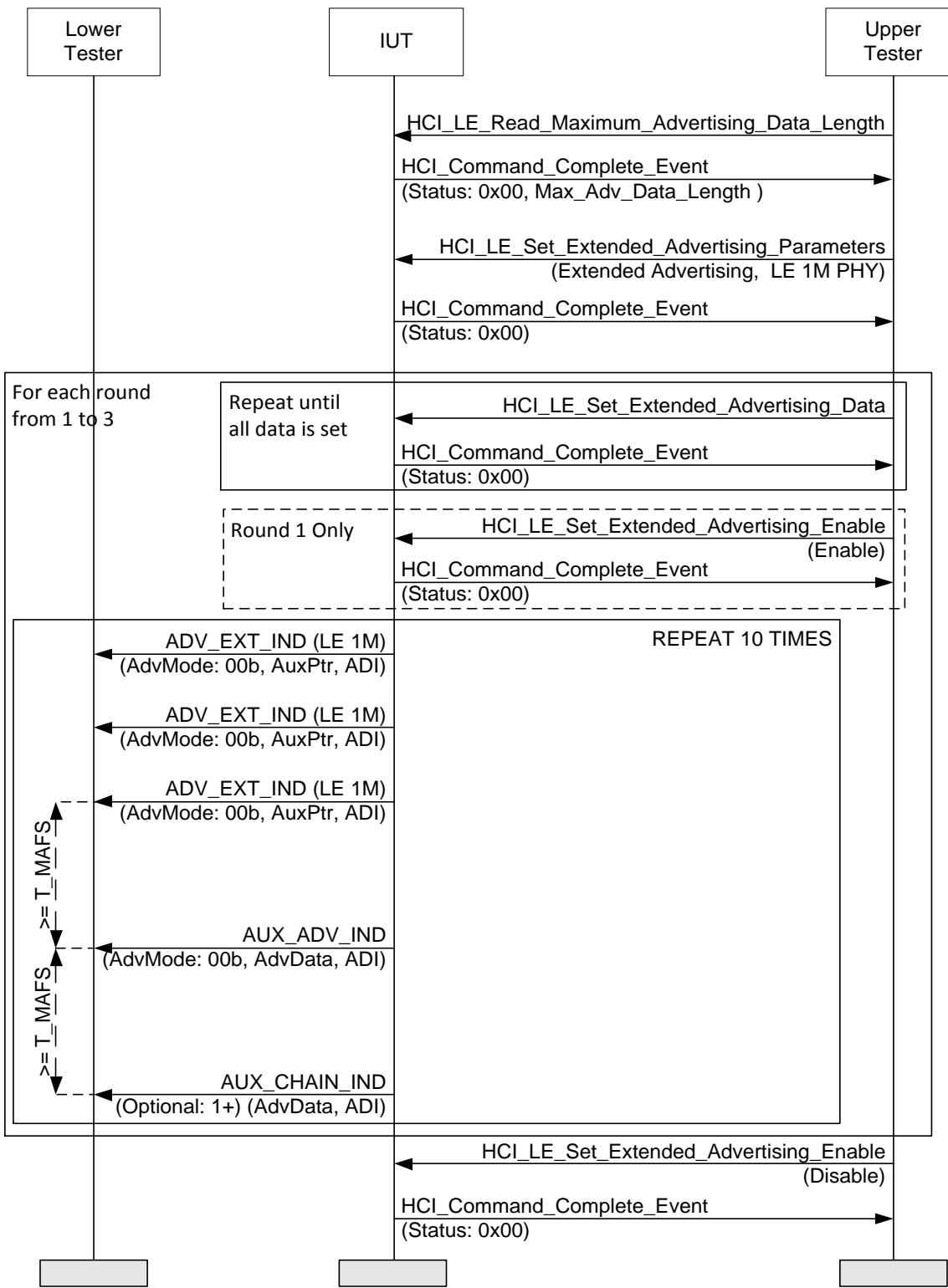


Figure 4.53: LL/DDI/ADV/BV-27-C [Extended Advertising, Host Modifying Data and ADI]

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



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

For each round from 1 to 3 based on [Table 4.9](#):

3. If the Data Length listed in [Table 4.9](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 4, otherwise skip to Step 10.
4. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT with values according to [Table 4.9](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
5. On Round 1 only the Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
6. The Lower Tester expects an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present. The ADI field shall be present and contain the Advertising Set ID (SID) used by the Upper Tester in Step 3 and an Advertising Data ID.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set to 00b and an ADI field matching the ADI field of the ADV_EXT_IND in Step 6. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 4, it shall include an AuxPtr field.
8. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in Step 4. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received.
9. Except for the first advertisement in round 1, the Lower Tester compares the data in the AUX_ADV_IND and any AUX_CHAIN_IND PDUs with that from the previous advertisement (the data shall be concatenated together and the boundaries between PDUs ignored). If the data is not the same but the Advertising Data ID field has not changed, a Fail Verdict is recorded.
10. Repeat steps 6–9 10 times.
11. Repeat steps 3–10 for each Round shown in [Table 4.9](#).
12. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.

Round	HCI_LE_Set_Extended_Advertising_Data (Step 4)	
	Data Length	
1	Maximum_Advertising_Data_Length	
2	1	
3	251	

Table 4.9: Payload contents for each case variation.



- Expected Outcome

Pass Verdict

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

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

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel. The ADI field shall be present and contain the Advertising Set ID (SID) used by the Upper Tester in Step 2 and an Advertising Data ID
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- The Lower Tester receives all the data sent in step 4, and no further data, in the AUX_ADV_IND PDU and zero or more AUX_CHAIN_IND PDUs when only reading data from PDUs with a new DID. The DID shall change every time the data advertised changes.
- The Advertising Data ID changes whenever the data changes.

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

- Test Purpose

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

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

- Reference

[10] 4.4.2.2.2, 2.3.4.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

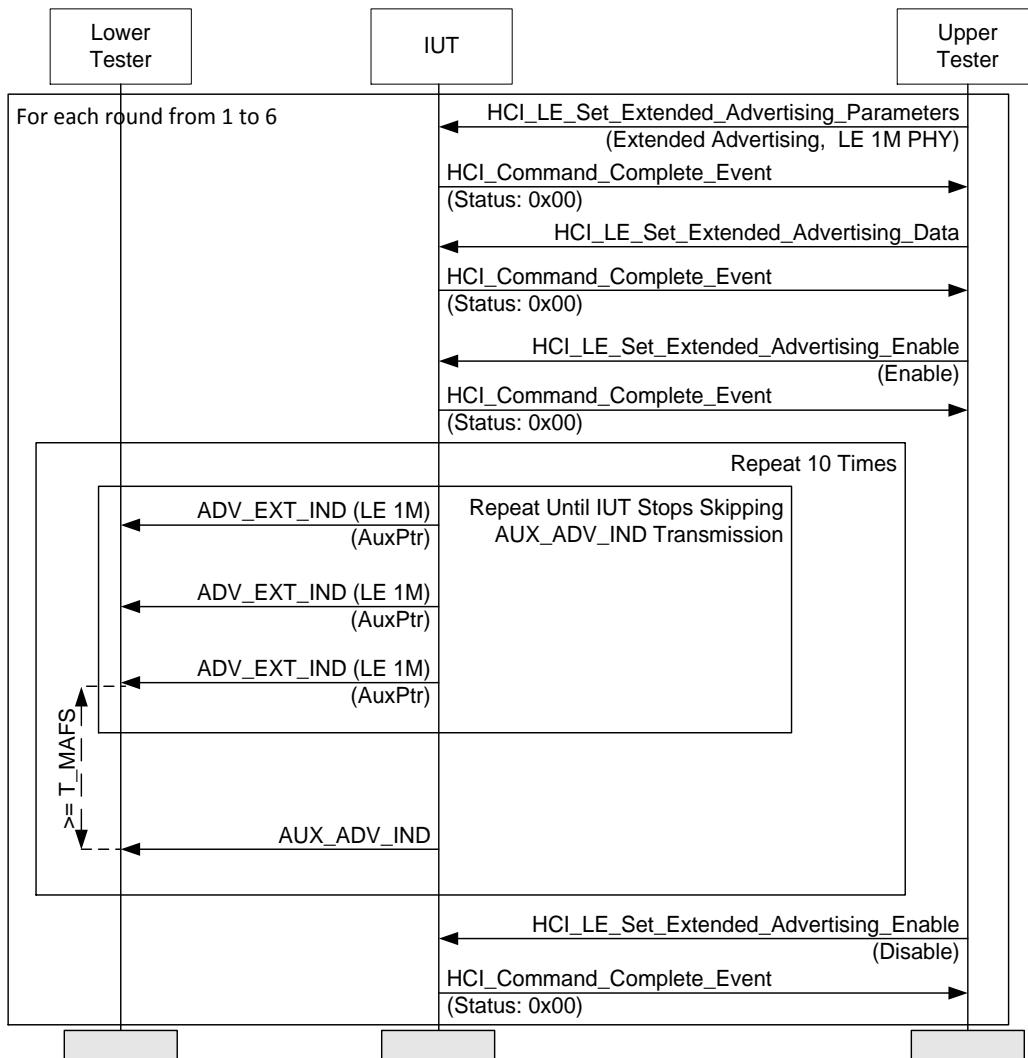


Figure 4.54: LL/DDI/ADV/BV-28-C [Extended Advertising, Overlapping Extended Advertising Events]

For each round from 1 to 6 based on [Table 4.10](#):

- The Upper Tester sends an **HCI_LE_Set_Extended_Advertising_Parameters** command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties shall be set according to [Table 4.10](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M). Secondary_Advertising_Max_Skip shall be set according to [Table 4.10](#).
- The Upper Tester sends an **HCI_LE_Set_Extended_Advertising_Data** command to the IUT with length 1 and using a random octet from 1 to 255 as the payload.
- The Upper Tester enables advertising using the **HCI_LE_Set_Extended_Advertising_Enable** command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).
- The Lower Tester expects advertising events consisting of **ADV_EXT_IND** packets from the IUT with the AuxPtr Extended Header field present and the AdvMode set according to expected properties in [Table 4.10](#).

5. The Lower Tester utilizes the AuxPtr's Aux Offset and Offset Units to calculate the expected time when the IUT will send an AUX_ADV_IND PDU on the secondary advertising channel.
6. Repeat Steps 4–5 until the target time from Step 5 is reached, recalculating the target time each round, expecting all target times to be within one Offset Unit of each other.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in [Table 4.10](#).
8. The Lower Tester records the total number of advertising events in which the IUT skipped sending an AUX_ADV_IND PDU.
9. Repeat Steps 4–8 the number of times specified in Table 4.9.
10. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
11. Repeat Steps 1–10 for each Round shown in [Table 4.10](#).

Round	Advertising_Event_Properties (Step 1)	Secondary_Advertising_Max_Skip (Step 1)	Repeat count (Step 9)
1	0x0000	0x01	100
2	0x0000	0x0F	50
3	0x0000	0xFF	10
4	0x0001	0x08	50
5	0x0004	0x08	50
6	0x0005	0x08	50

Table 4.10: Advertising properties for each case variation.

- Expected Outcome

Pass Verdict

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

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

[4.2.2.27 LL/DDI/ADV/BV-29-C \[Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY\]](#)

- Test Purpose

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

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



- Reference

[10] 4.4.2.6

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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

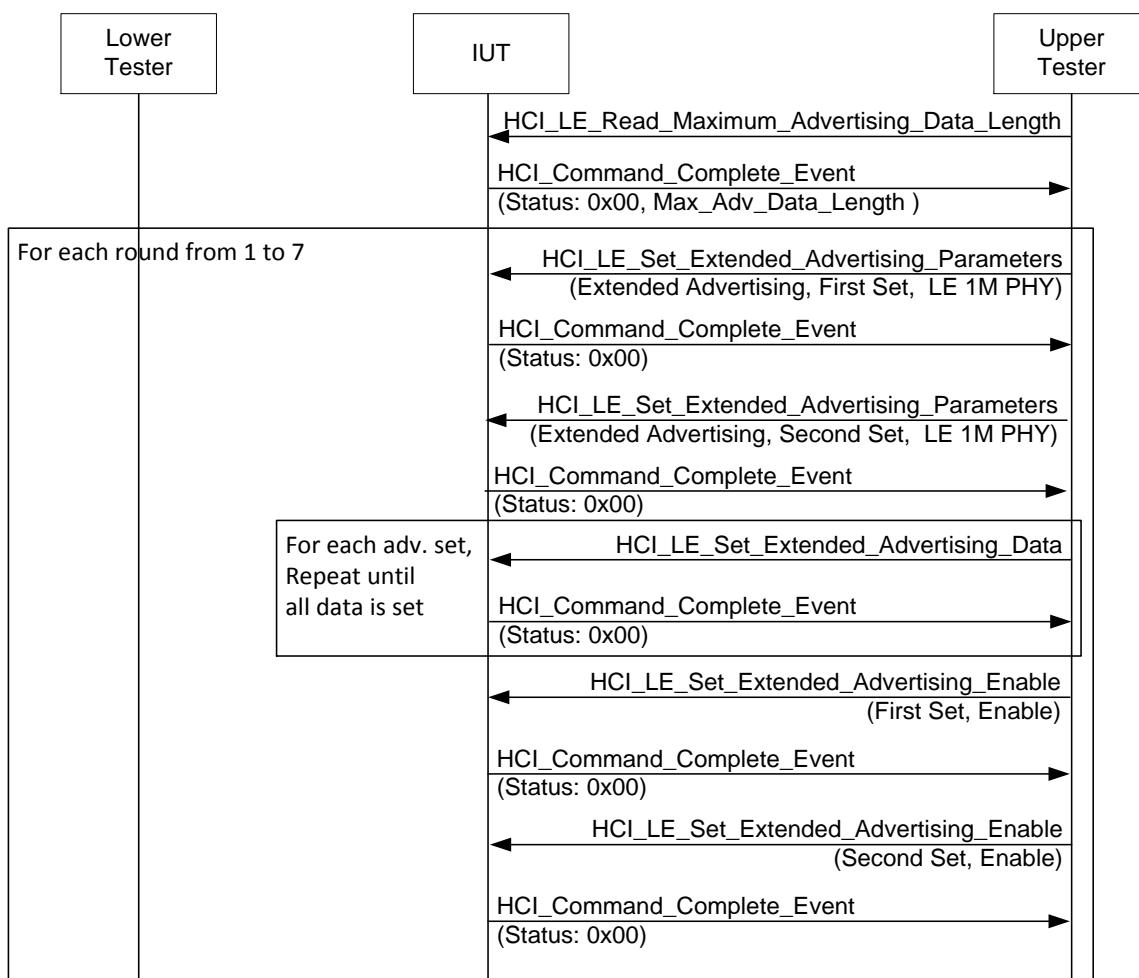


Figure 4.55: LL/DDI/ADV/BV-29-C [Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY] – Part A



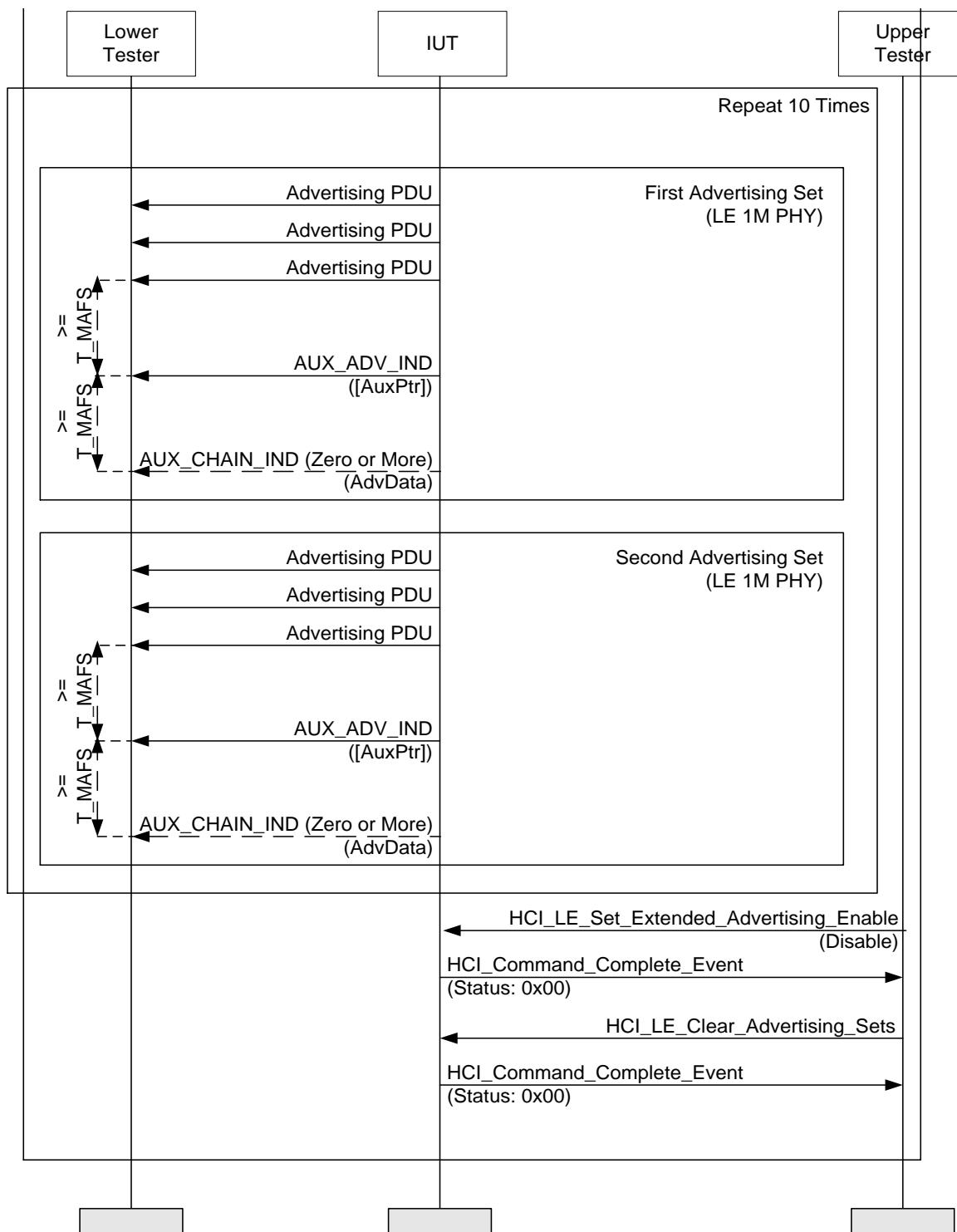


Figure 4.56: LL/DDI/ADV/BV-29-C [Extended Advertising, Multiple Sets, Single PHY, LE 1M PHY] – Part B

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

For each round from 1 to 7 based on [Table 4.11](#):

2. If the Data Length listed in [Table 4.11](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 12.



3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in [Table 4.11](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in [Table 4.11](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to [Table 4.11](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).
7. The Lower Tester expects interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in [Table 4.11](#) for each set.
8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in [Table 4.11](#) for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 5 for the set (if any), it shall include an AuxPtr field.
9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in Step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.
10. Repeat steps 7–9 10 times.
11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.
12. The Upper Tester clears the advertising sets using one HCI_LE_Clear_Advertising_Sets command.
13. Repeat steps 2–12 for each Round shown in [Table 4.11](#).



Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set	HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set	HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set		HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set	
	Advertising_Event_Properties	Advertising_Event_Properties	Data Length		Data Length	
1	0x0000	0x0001	0		0	
2	0x0001	0x0000	0		191	
3	0x0000	0x0005	191		0	
4	0x0000	0x0000	191		191	
5	0x0000	0x0004	251		251	
6	0x0000	0x0001	253		0	
7	0x0000	0x0000	474		0	

Table 4.11: Payload contents for each case variation

- Expected Outcome

Pass Verdict

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

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

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

4.2.2.28 LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY]

- Test Purpose

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

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

- Reference

[10] 4.4.2.6



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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

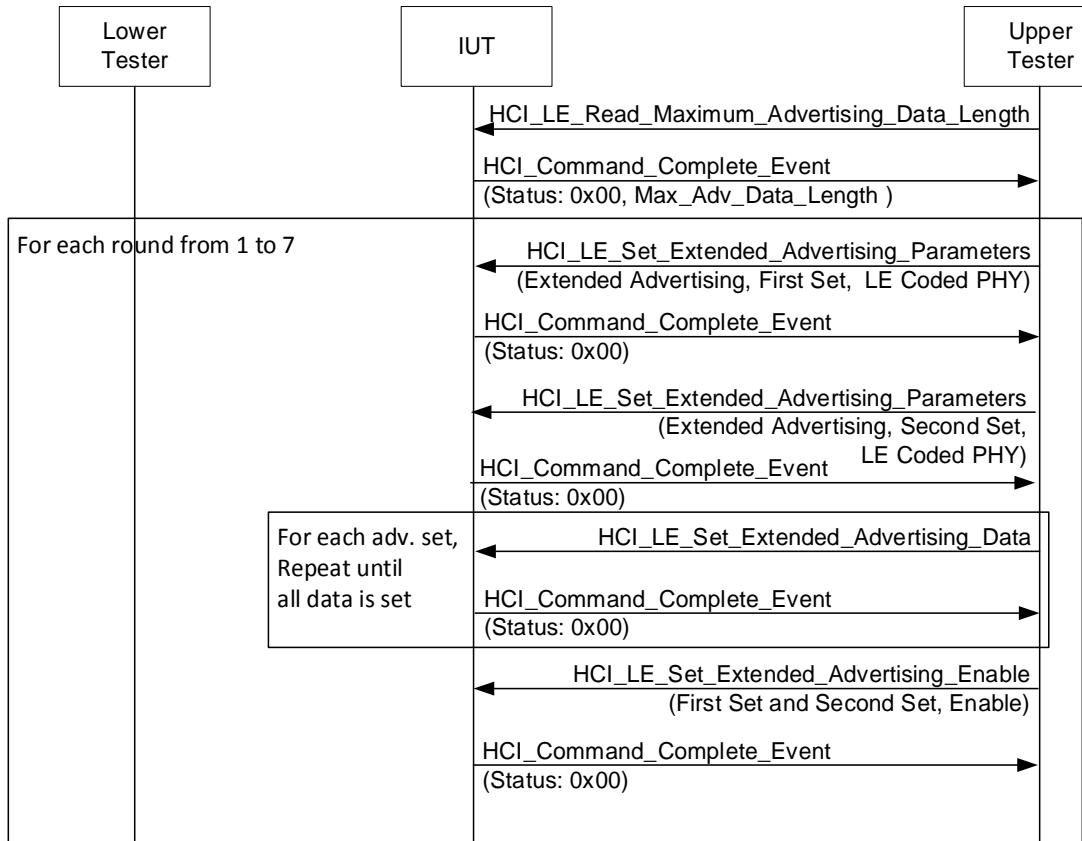


Figure 4.57: LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY] – Part A



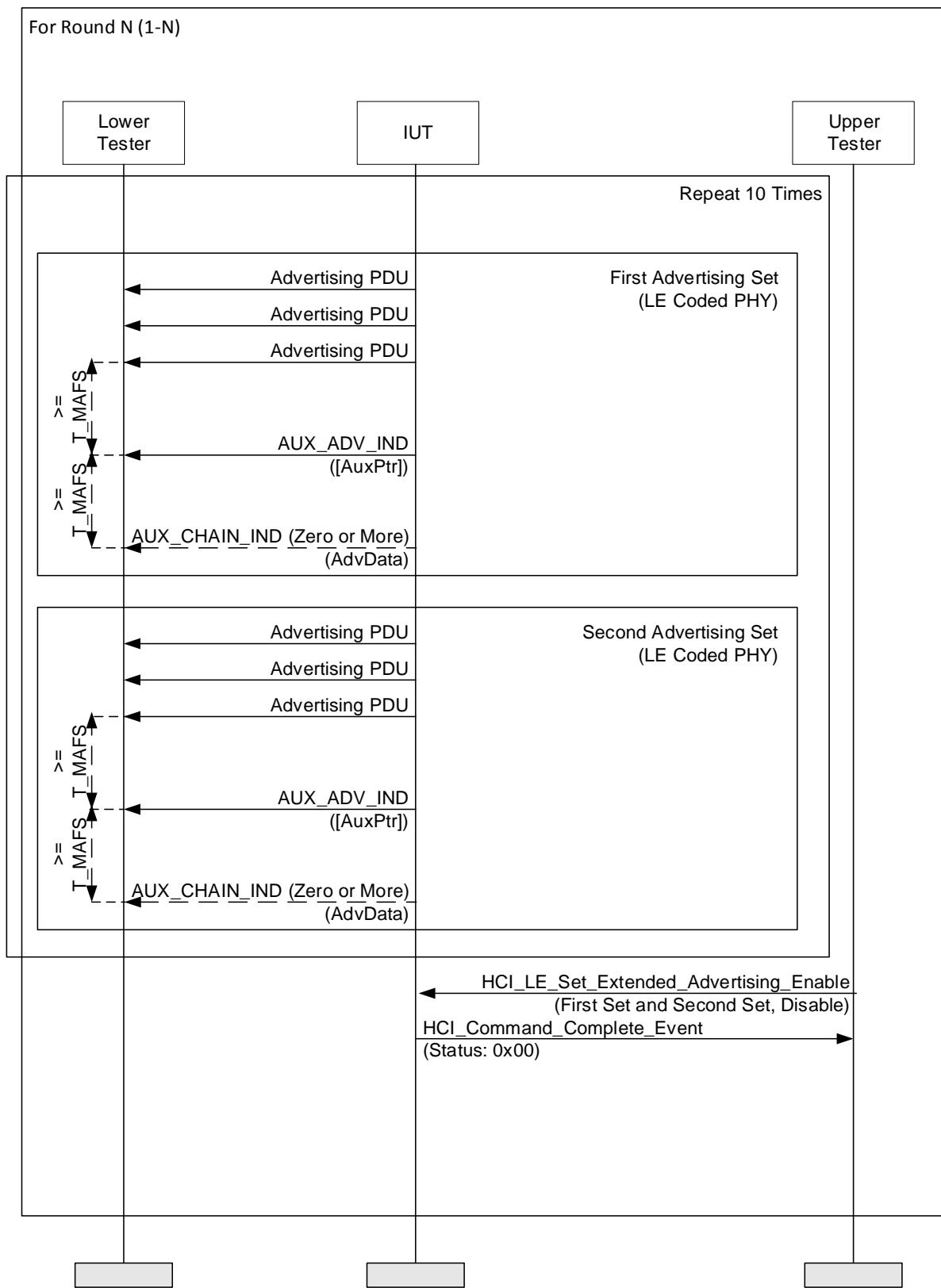


Figure 4.58: LL/DDI/ADV/BV-30-C [Extended Advertising, Multiple Sets, Single PHY, LE Coded PHY] – Part B

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

For each round from 1 to 7 Based on [Table 4.12](#):

2. If the Data Length listed in [Table 4.12](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 12.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in [Table 4.12](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in [Table 4.12](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to [Table 4.12](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).
7. The Lower Tester expects interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in [Table 4.12](#) for each set using the LE Coded PHY.
8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in [Table 4.12](#) for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 5 for the set (if any), it shall include an AuxPtr field.
9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in Step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.
10. Repeat steps 7-9 10 times.
11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.
12. Repeat steps 2-11 for each Round shown in [Table 4.12](#).



Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set	HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set	HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set		HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set	
	Advertising_Event_Properties	Advertising_Event_Properties	Data Length		Data Length	
1	0x0000	0x0001	0		0	
2	0x0001	0x0000	0		191	
3	0x0000	0x0005	191		0	
4	0x0000	0x0000	191		191	
5	0x0000	0x0004	251		251	
6	0x0000	0x0001	253		0	
7	0x0000	0x0000	474		0	

Table 4.12: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

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

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

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

- Test Purpose

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

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

- Reference

[10] 4.4.2.6



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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

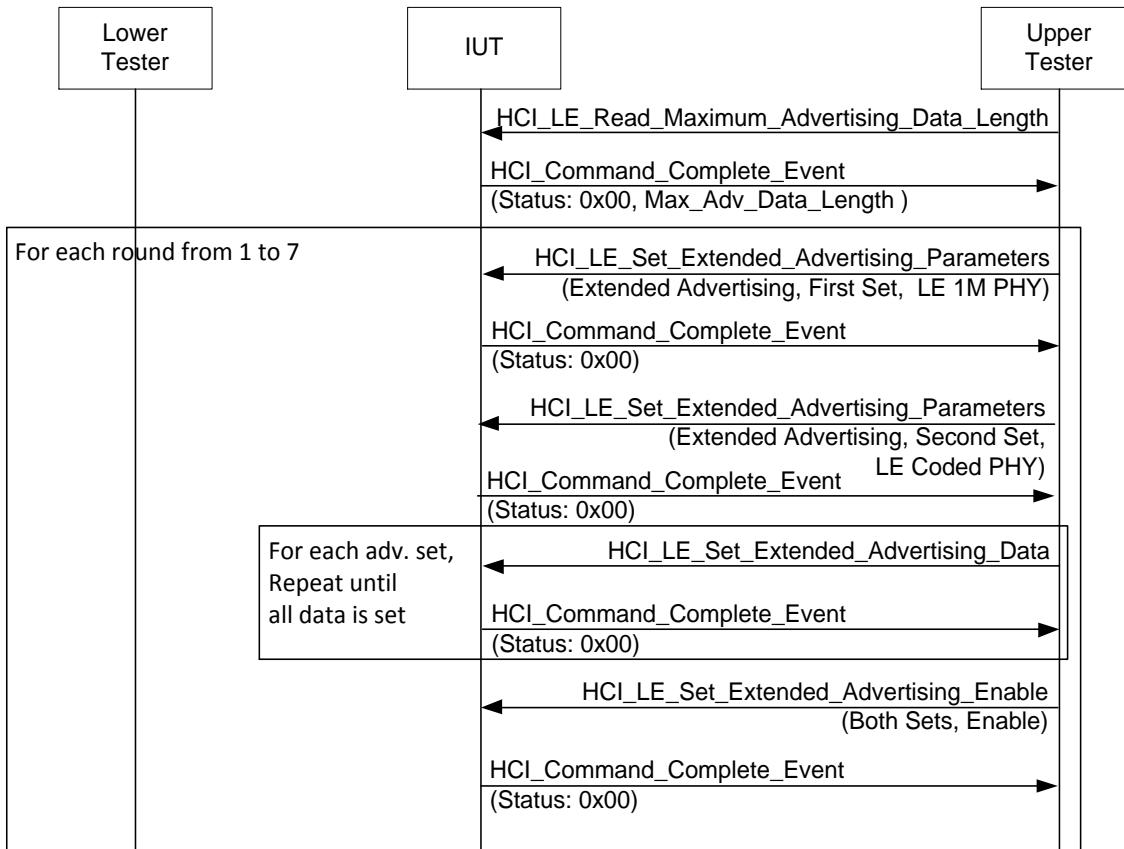


Figure 4.59: LL/DDI/ADV/BV-31-C [Extended Advertising, Multiple Sets, Multiple PHYs, LE 1M and LE Coded PHYs] – Part A



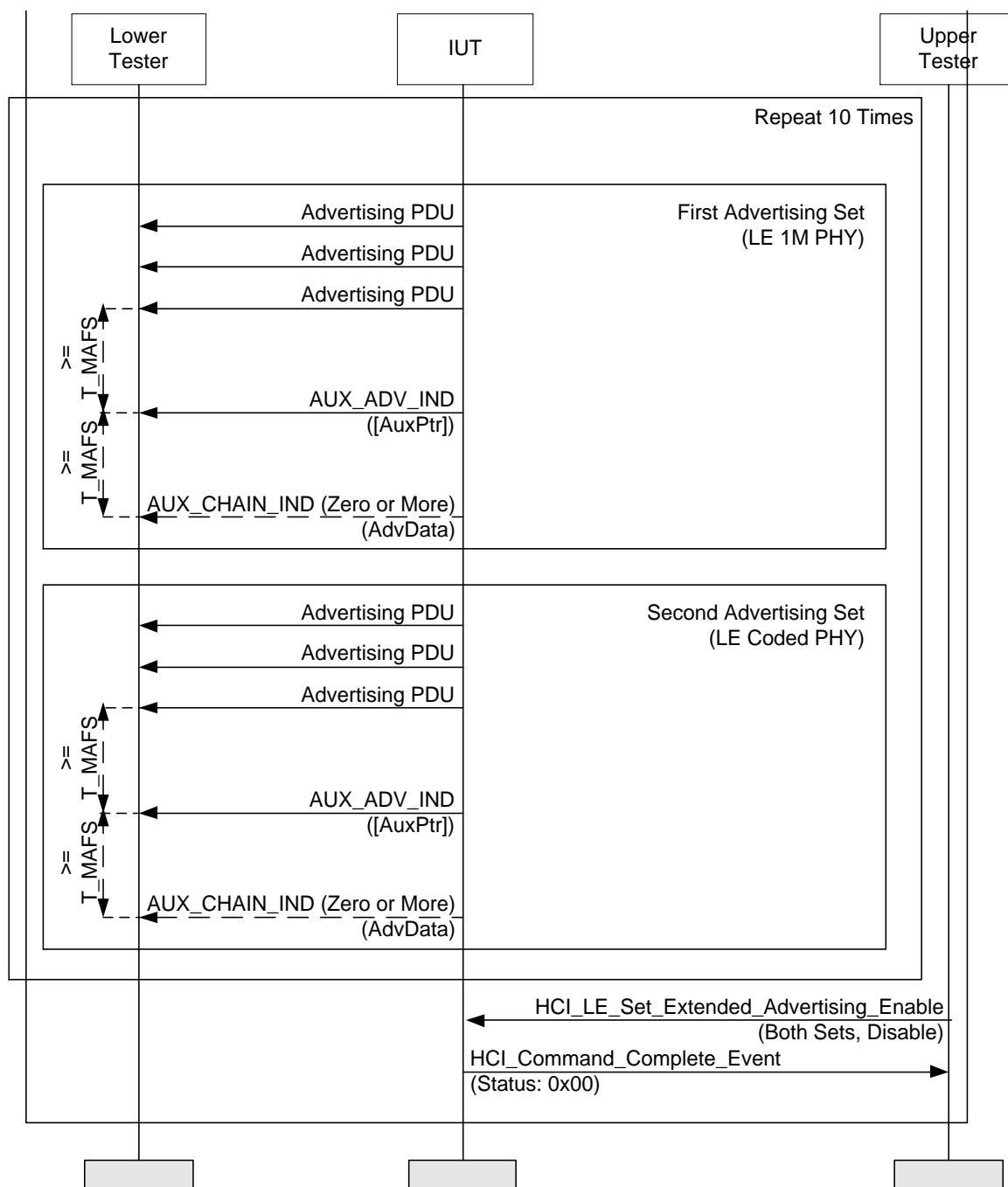


Figure 4.60: LL/DDI/ADV/BV-31-C [Extended Advertising, Multiple Sets, Multiple PHYs, LE 1M and LE Coded PHYs] – Part B

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

For each round from 1 to 7 based on [Table 4.13](#):

2. If the Data Length listed in [Table 4.13](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 12.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the



minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in [Table 4.13](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in [Table 4.13](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x03 (LE Coded). The Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
5. The Upper Tester sends one or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to [Table 4.13](#) and using random octets from 1 to 255 as the payload. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).
7. The Lower Tester expects interlaced advertising events for each set, consisting of ADV_EXT_IND packets from the IUT with the AdvMode set according to expected properties in [Table 4.13](#). The first set shall use the LE 1M PHY and the second set shall use the LE Coded PHY.
8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the appropriate PHY with the AdvMode field set according to expected properties in [Table 4.13](#) for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 5 for the set (if any), it shall include an AuxPtr field.
9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.
10. Repeat steps 7-9 10 times.
11. The Upper Tester both disables advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.
12. Repeat steps 2-11 for each Round shown in [Table 4.13](#).



Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set	HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set	HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set		HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set	
	Advertising_Event_Properties	Advertising_Event_Properties	Data Length		Data Length	
1	0x0000	0x0001	0		0	
2	0x0001	0x0000	0		191	
3	0x0000	0x0005	191		0	
4	0x0000	0x0000	191		191	
5	0x0000	0x0004	251		251	
6	0x0000	0x0001	253		0	
7	0x0000	0x0000	474		0	

Table 4.13: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

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

- The IUT interlaces multiple advertising events for each set. Each set consists of ADV_EXT_IND PDUs with an optional AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel. The first set shall use the LE 1M PHY and the second set shall use the LE Coded PHY
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- For each set, if advertising data was specified for the round, the Lower Tester receives all the data sent in step 5, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set using the appropriate PHY. If no data was specified, any AUX_ADV_IND and AUX_CHAIN_IND PDUs shall not contain any data.

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

- Test Purpose

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

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

- Reference

[10] 4.4.2.6



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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

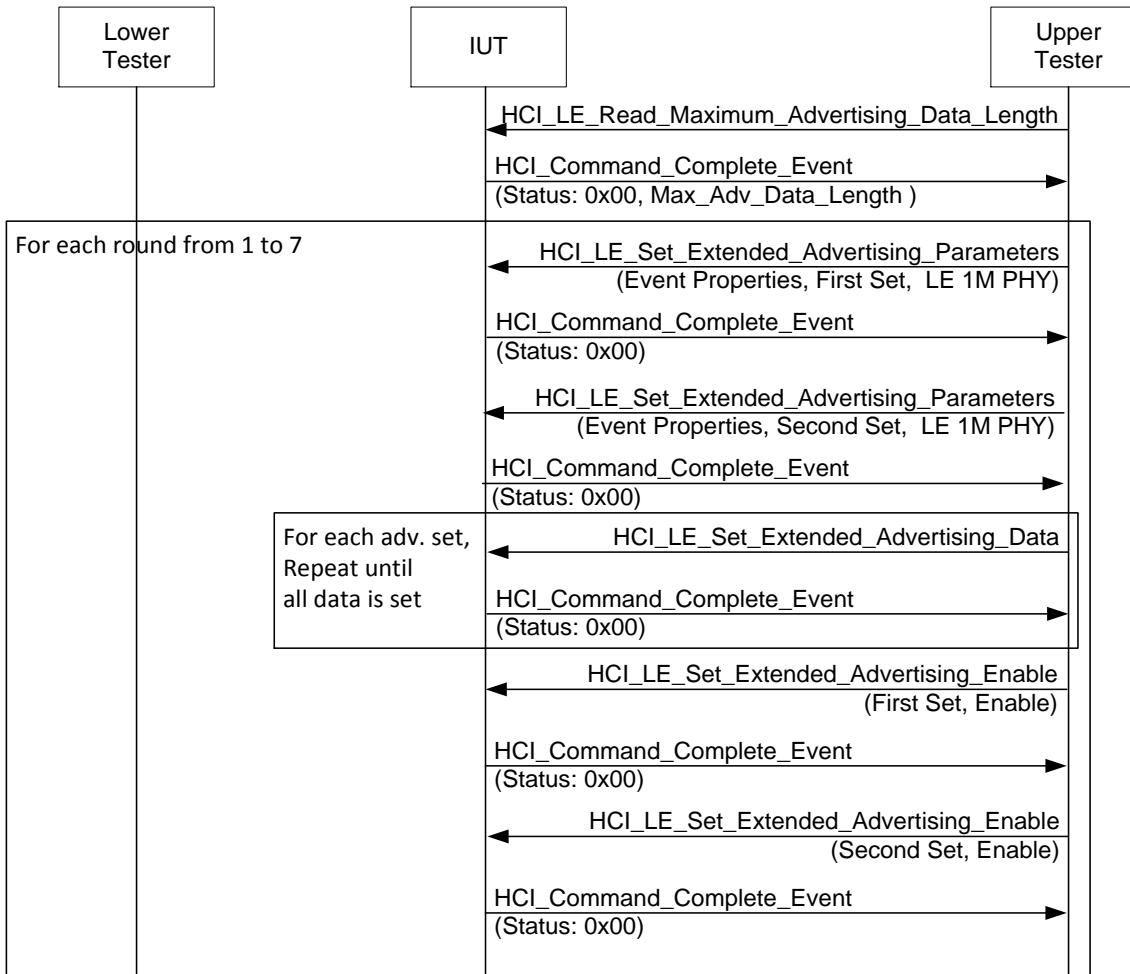


Figure 4.61: LL/DDI/ADV/BV-32-C [Extended Advertising, Multiple Sets, Legacy and Extended] – Part A

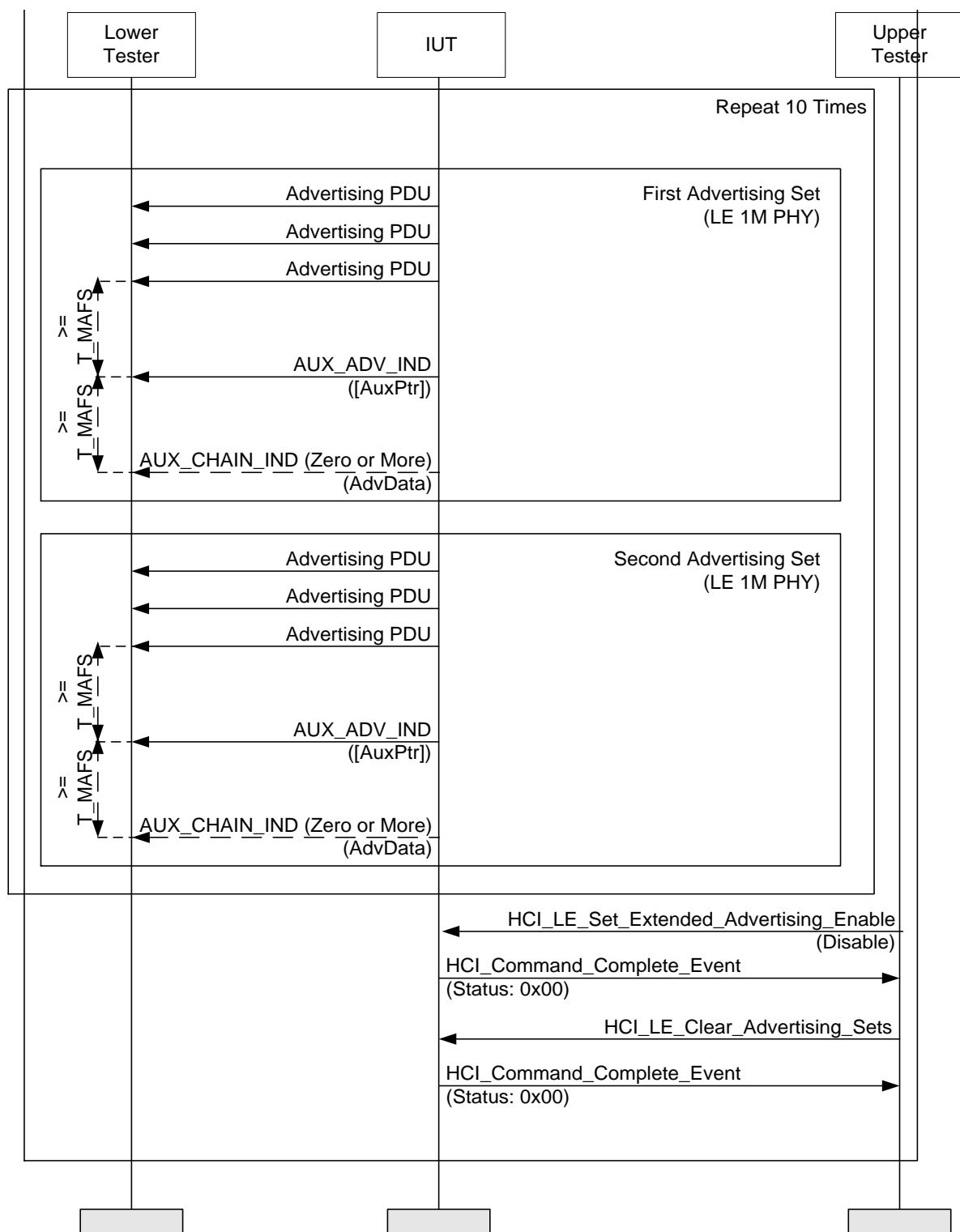


Figure 4.62: LL/DDI/ADV/BV-32-C [Extended Advertising, Multiple Sets, Legacy and Extended] – Part B

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



For each round from 1 to 7 based on [Table 4.14](#):

2. If the Data Length listed in [Table 4.14](#) for the current Round is less than or equal to the Maximum_Advertising_Data_Length proceed to Step 3, otherwise skip to Step 12.
3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the First Set column in [Table 4.14](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set according to the Second Set column in [Table 4.14](#). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
5. The Upper Tester sends zero or more HCI_LE_Set_Extended_Advertising_Data commands to the IUT for each advertising set, with values according to [Table 4.14](#) and using random octets from 1 to 255 as the payload. If the Data Length is zero for a given set, the Upper Tester shall do nothing for that set. If the Data Length is greater than 251 for a given set, the Upper Tester shall send multiple commands using one Operation 0x01 (First fragment) command, followed by zero or more Operation 0x00 (Intermediate Fragment) commands, and a final Operation 0x02 (Last fragment) command. Otherwise the Upper Tester shall send a single command using Operation 0x03 (Complete Data).
6. The Upper Tester enables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, if the Advertising_Event_Properties parameter was set to 0x1D in step 3 or 4, then Duration[i] is set to 0x0080 (1280ms); otherwise the Duration[i] parameter is set to 0x0000 (No Advertising Duration).
7. The Lower Tester expects interlaced advertising events for each set, consisting of either legacy PDUs or ADV_EXT_IND PDUs from the IUT with the AdvMode set according to expected properties in [Table 4.14](#) for each set.
8. If an AuxPtr field is present, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY with the AdvMode field set according to expected properties in [Table 4.14](#) for the set. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 5 for the set (if any), it shall include an AuxPtr field.
9. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b and containing additional data submitted in Step 5 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.
10. Repeat steps 7-9 10 times.
11. The Upper Tester disables both advertising sets using one HCI_LE_Set_Extended_Advertising_Enable command.
12. The Upper Tester clears the advertising sets using one HCI_LE_Clear_Advertising_Sets command.
13. Repeat steps 2-11 for each Round shown in [Table 4.14](#).



Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 3) First Advertising Set	HCI_LE_Set_Extended_Advertising_Parameters (Step 4) Second Advertising Set	HCI_LE_Set_Extended_Advertising_Data (Step 5) First Advertising Set		HCI_LE_Set_Extended_Advertising_Data (Step 5) Second Advertising Set	
	Advertising_Event_Properties	Advertising_Event_Properties	Data Length		Data Length	
1	0x0000	0x0010	0		0	
2	0x0013	0x0000	0		191	
3	0x0000	0x0013	191		31	
4	0x0013	0x0010	31		31	
5	0x0015	0x0004	0		251	
6	0x0000	0x001D	253		0	
7	0x0000	0x0012	474		0	

Table 4.14: Payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

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

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

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

- Test Purpose

Tests that an advertiser IUT can support multiple periodic advertising sets on the same PHY or different PHYs.

The Lower Tester observes the event timing and packet contents on the advertising channels in use. The Lower Tester confirms that Channel Selection Algorithm #2 is utilized for the periodic advertisements.

- Reference

[10] 4.4.2.2.2, 2.3.4.5



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

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

- Test Procedure

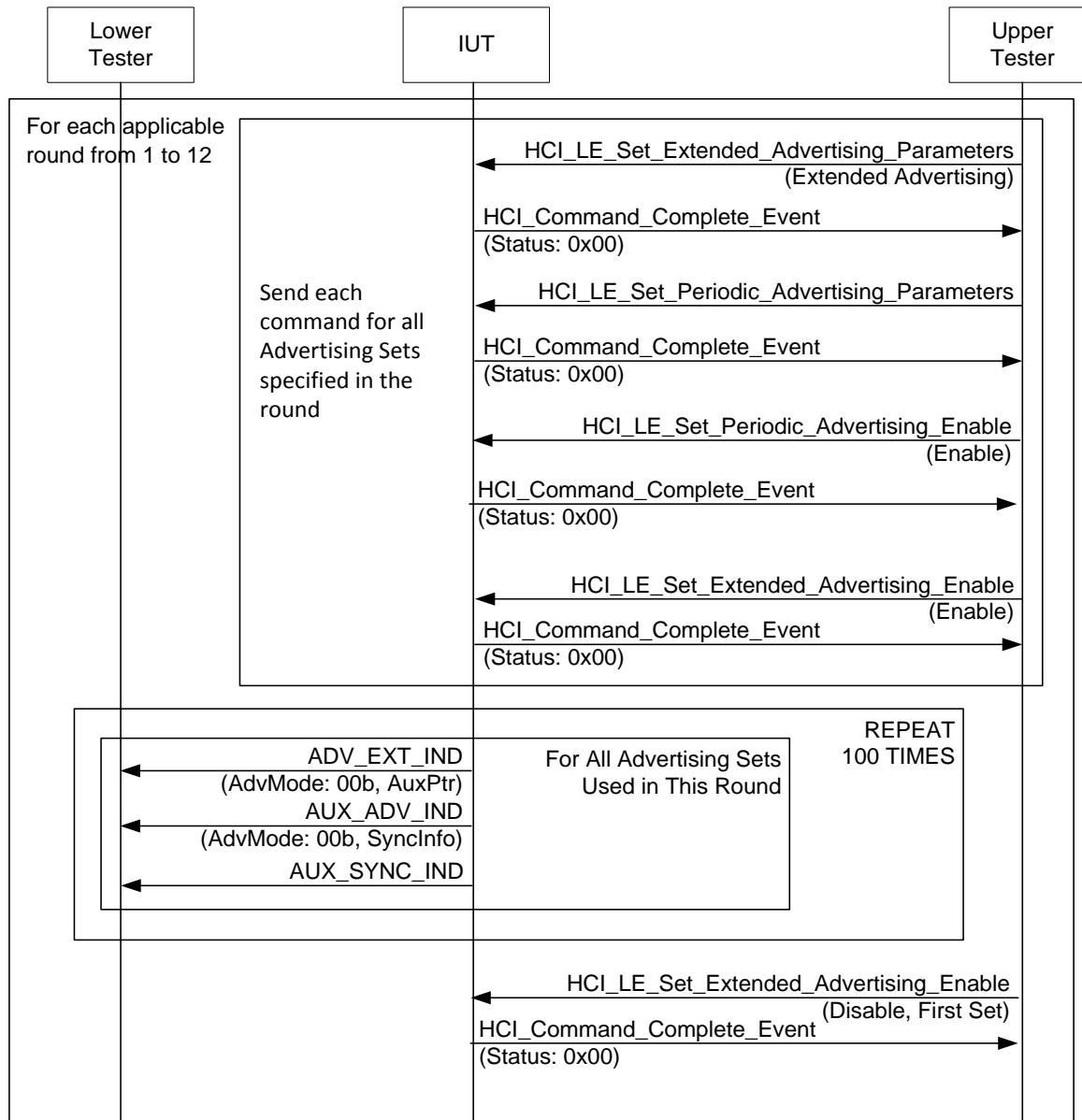


Figure 4.63: LL/DDI/ADV/BV-33-C [Extended Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)] – Part A



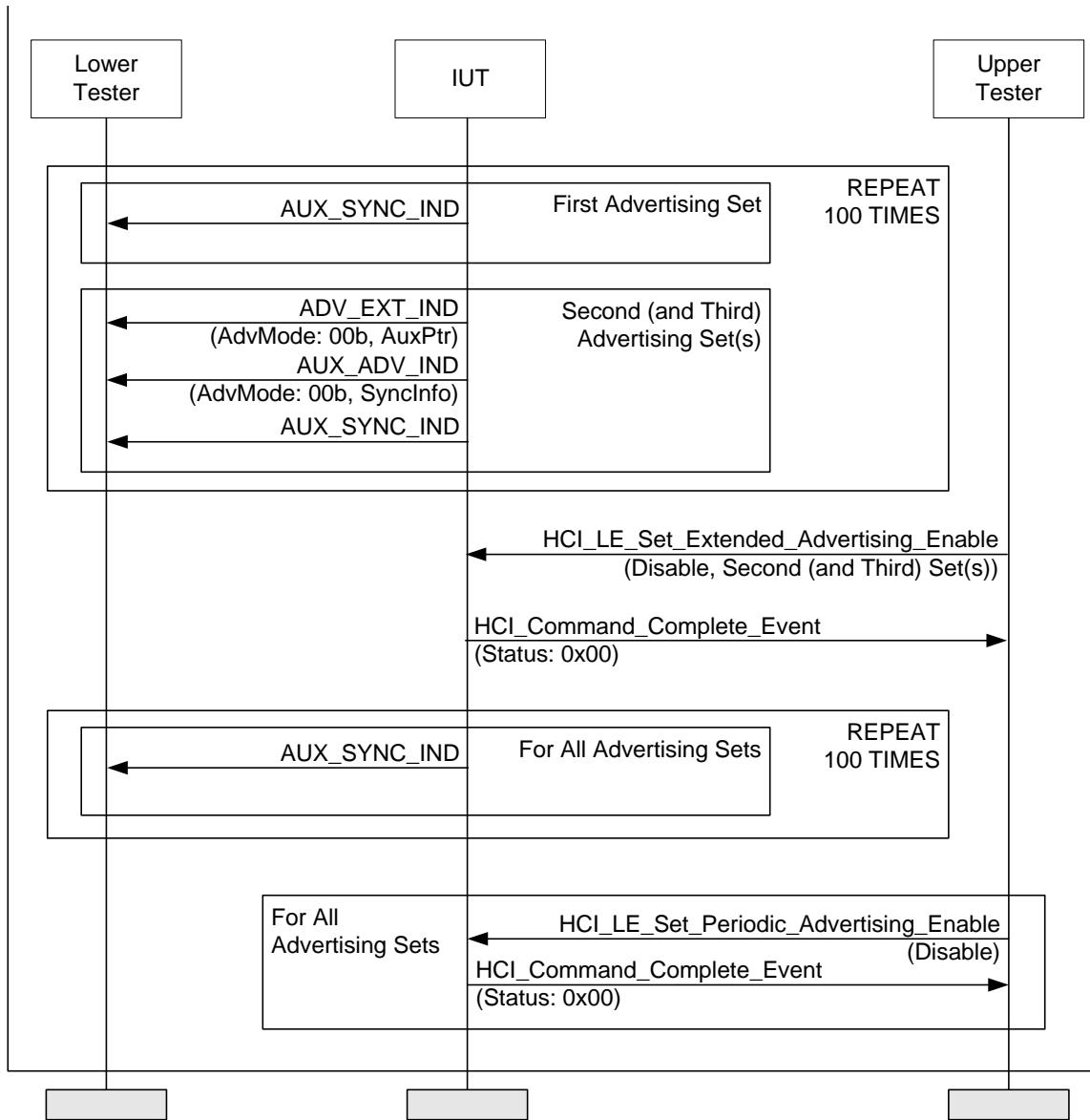


Figure 4.64: LL/DDI/ADV/BV-33-C [Extended Advertising, Periodic Advertising, Multiple Sets, Multiple PHYs (All Supported PHYs)] – Part B

For each applicable round in Table 4.15 below:

1. The Upper Tester sends HCI_LE_Set_Extended_Advertising_Parameters commands to the IUT to create the specified number of advertising sets using all supported advertising channels and 50ms advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The PHY for primary and secondary advertising channels shall be as specified in the table. The Upper Tester expects an HCI_Command_Complete event from the IUT for each command with Status set to 0x00 (Success).
 2. For each set, the Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels. Periodic_Advertising_Properties parameter shall be set to 0x0000. For each round, the advertising intervals for the first, second and third sets shall be set to, respectively, 15ms, 20ms and 22.5ms. The Upper Tester expects an

- HCI_Command_Complete event from the IUT for each command with Status set to 0x00 (Success).
3. For each set, the Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (Periodic Advertising).
 4. The Upper Tester enables each advertising set using one HCI_LE_Set_Extended_Advertising_Enable command. For each set i, the Duration[i] parameter is set to 0x0000 (No Advertising Duration).
 5. The Lower Tester scans for each advertising PHY supported by the IUT. For each set, the Lower Tester expects interlaced advertising events from the IUT using the Primary Advertising PHY for which that set was configured in Step 1, consisting of ADV_EXT_IND packets with AdvMode set to 00b with the AuxPtr Extended Header field present.
 6. For each set, the Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the Secondary Advertising PHY for which that set was configured in Step 1, with the AdvMode field set to 00b and the SyncInfo Extended Header fields present.
 7. Repeat Steps 5-6 100 times. At the same time, for each set, the Lower Tester utilizes the SyncInfo field to listen for AUX_SYNC_IND PDUs on the secondary advertising using the indices selected by the LE Channel Selection Algorithm #2 using the Secondary Advertising PHY for which that set was configured in Step 1 and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDUs shall have the AdvMode field set to 00b.
 8. The Upper Tester disables extended advertising for the first set using the HCI_LE_Set_Extended_Advertising_Enable command but maintains synchronization with the IUT's periodic advertising.
 9. The Lower Tester confirms that periodic advertising continues for the first set and that both extended advertising and periodic advertising continue for the remaining set(s) by repeating steps 5-7, except that in steps 5-6, the Lower Tester only expects advertising PDUs for the remaining set(s).
 10. The Upper Tester disables extended advertising for the remaining set(s) using the HCI_LE_Set_Extended_Advertising_Enable command but maintains synchronization with the IUT's periodic advertising.
 11. For both sets on each supported advertising PHY, the Lower Tester maintains synchronization to periodic advertising and expects to receive at least 100 AUX_SYNC_IND PDUs on the secondary advertising using the indices selected by the LE Channel Selection Algorithm #2 using the Secondary Advertising PHY for which that set was configured in Step 1
 12. The Upper Tester disables periodic advertising on all sets using the HCI_LE_Set_Periodic_Advertising_Enable command.

Table 4.15 specifies the rounds to be carried out based on the supported PHYs and the maximum number of supported advertising sets. Round 1 is always carried out; the remaining rounds are only carried out if a PHY other than LE 1M is supported.

Round	Supported PHYs and max sets				First set		Second set		Third set	
	2M ≥ 2	Coded ≥ 2	Both 2	Both ≥ 3	Primary PHY	Secon'y PHY	Primary PHY	Secon'y PHY	Primary PHY	Secon'y PHY
1	•	•	•	•	1M	1M	1M	1M		
2	•		•	•	1M	2M	1M	2M		
3		•	•	•	Coded	Coded	Coded	Coded		
4	•		•		1M	1M	1M	2M		



Round	Supported PHYs and max sets				First set		Second set		Third set	
	2M ≥ 2	Coded ≥ 2	Both 2	Both ≥ 3	Primary PHY	Secon'y PHY	Primary PHY	Secon'y PHY	Primary PHY	Secon'y PHY
5	•		•		1M	2M	1M	1M		
6		•	•		1M	1M	Coded	Coded		
7		•	•		Coded	Coded	1M	1M		
8			•		1M	2M	Coded	Coded		
9			•		Coded	Coded	1M	2M		
10				•	1M	1M	1M	2M	Coded	Coded
11				•	1M	2M	Coded	Coded	1M	1M
12				•	Coded	Coded	1M	1M	1M	2M

Table 4.15: Specification of rounds for LL/DDI/ADV/BV-33-C

- Expected Outcome

Pass Verdict

For each advertising set specified in the test procedure, the following condition shall occur:

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel using the advertising PHY for which that set was configured.
- The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.
- The IUT sends AUX_SYNC_IND PDUs with correct timings on the secondary advertising channel using indices selected by the Channel Selection Algorithm #2.

Inconclusive Verdict

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

4.2.2.32 LL/DDI/ADV/BV-34-C [Extended Advertising, TX Power with RF Path Compensation]

- Test Purpose

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

- Reference

[10] 2.3.4.7, 4.4.2.6

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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



- Test Procedure

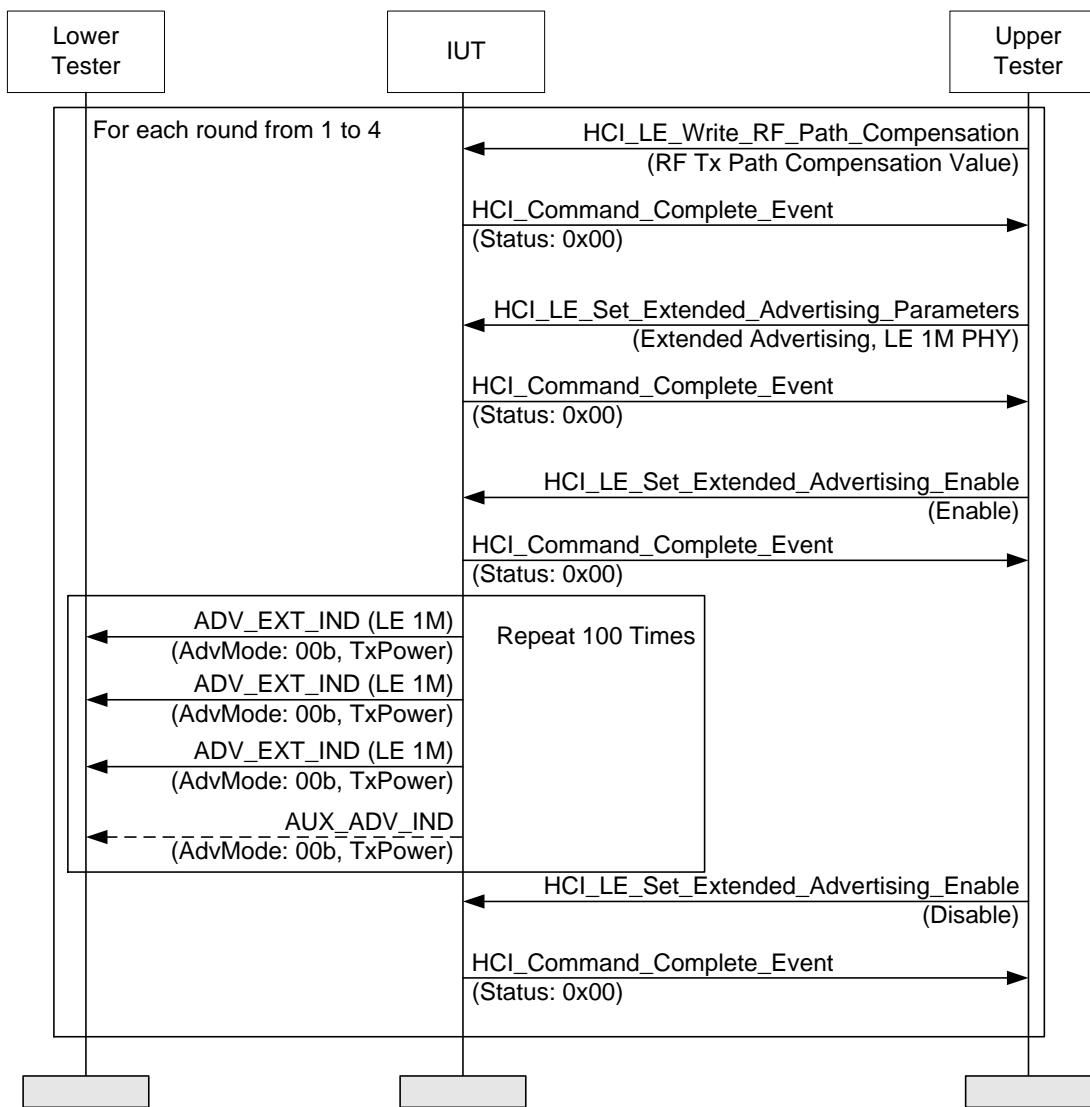


Figure 4.65: LL/DDI/ADV/BV-34-C [Extended Advertising, TX Power with RF Path Compensation]

For each round from 1 to 4 based on [Table 4.16](#):

1. The Upper Tester sends an **HCI_LE_Write_RF_Path_Compensation** command to the IUT. The **RF_Tx_Path_Compensation_Value** shall be set as shown in [Table 4.16](#) for the current Round.
2. The Upper Tester sends an **HCI_LE_Set_Extended_Advertising_Parameters** command to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. **Advertising_Event_Properties** parameter bit 6 (Include TxPower in the advertising PDU) shall be set and all other bits cleared. The **Own_Address_Type** shall be set to 0x00 (Public Device Address). The **Primary_Advertising_PHY** shall be set to 0x01 (LE 1M).
3. The Upper Tester enables advertising using the **HCI_Command_Complete_Event** (Status: 0x00) command. The **Duration[0]** parameter is set to 0x0000 (No Advertising Duration).
4. The Lower Tester expects the IUT to send **ADV_EXT_IND** on the first supported advertising channel. The **AdvMode** shall be set to 00b. If an **AuxPtr** field is present, the Lower Tester expects the IUT to send an **AUX_ADV_IND** on the specified secondary channel with the **AdvMode** set to 00b and that has a **TxPower** field present; otherwise the **ADV_EXT_IND** shall have a **TxPower**



- field present. The TxPower field shall contain a value adjusted for the RF_Tx_Path_Compensation_Value used in this round, relative to the unadjusted value received in Round 1.
5. Repeat Step 4 until 100 advertising events have been detected.
 6. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
 7. Repeat Steps 1–6 for each Round shown in [Table 4.16](#).

Round	HCI_LE_Write_RF_Path_Compensation (Step 1)
	RF Tx Path Compensation Value
1	0 dB
2	+5 dB
3	-5 dB
4	0 dB

Table 4.16: RF Tx Path Compensation Values.

- Expected Outcome

Pass Verdict

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

- The Lower Tester receives an ADV_EXT_IND with a Tx Power field.
- The Tx Power field contains a value adjusted for the RF_Tx_Path_Compensation_Value used relative to the unadjusted value received in Round 1.

[4.2.2.33 LL/DDI/ADV/BV-35-C \[Extended Advertising, Multiple Sets, Maximum Supported Sets\]](#)

- Test Purpose

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

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

- Reference

[\[10\] 4.4.2.6](#)

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

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



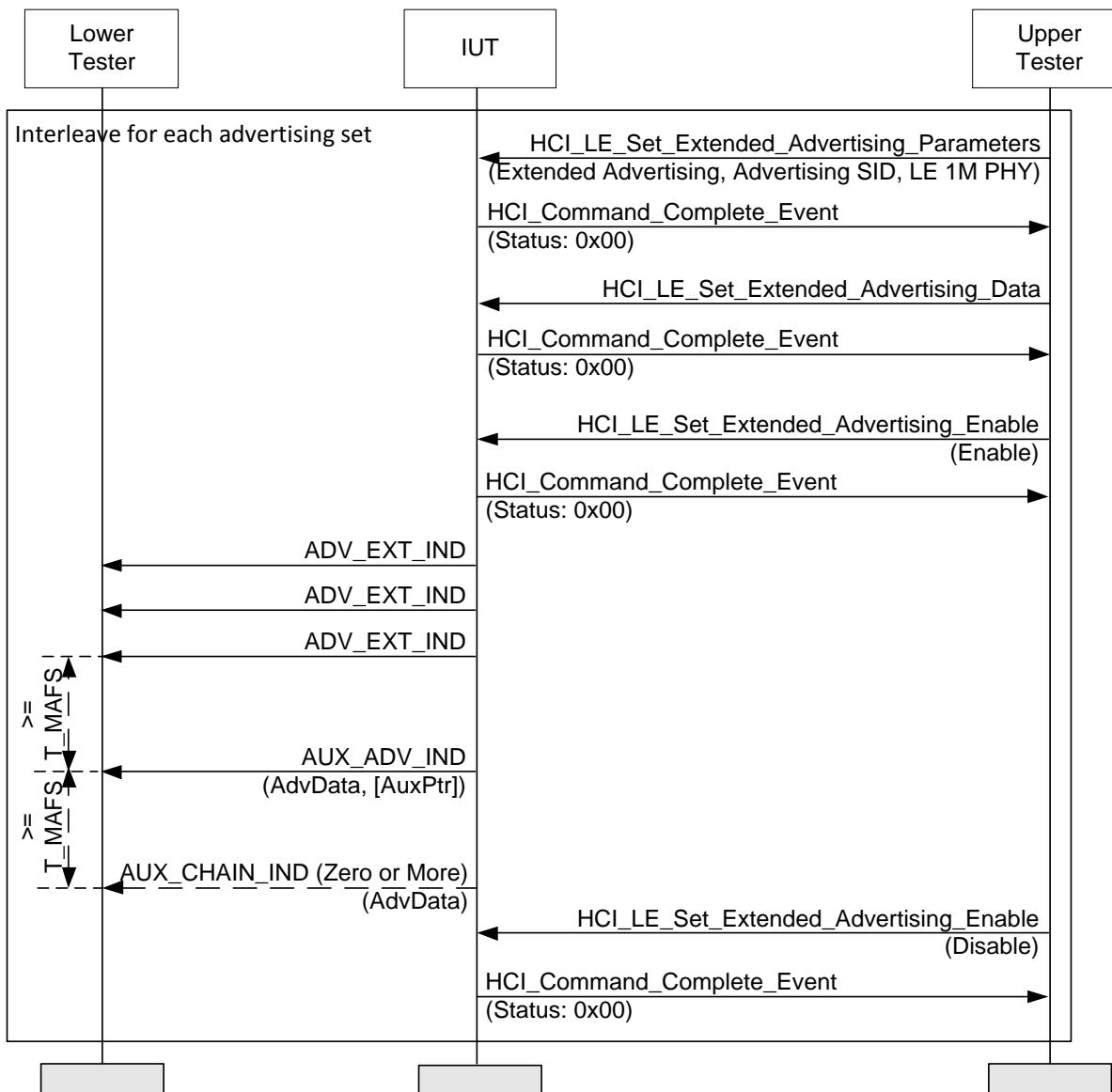


Figure 4.66: LL/DDI/ADV/BV-35-C [Extended Advertising, Multiple Sets, Maximum Supported Sets]

1. The Upper Tester sends TSPX_adv_sets_max HCI_LE_Set_Extended_Advertising_Parameters commands to the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals supported. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M). The Advertising_SID shall be set to a random value, [0x00 ... 0x0F], for each set.
 2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command to the IUT for each advertising set, with length 31 and using random octets from 1 to 255 as the payload.
 3. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command for each set. The Duration parameter is set to 0x0000 (No Advertising Duration).
 4. The Lower Tester expects interlaced advertising events for each set, consisting of ADV_EXT_IND PDUs from the IUT. The ADV_EXT_IND PDUs shall contain an AuxPtr field.
 5. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel using the LE 1M PHY. If the AUX_ADV_IND PDU does not contain all the data submitted in Step 2 for the set, it shall include an AuxPtr field.

6. If the AUX_ADV_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU containing additional data submitted in Step 2 for the set. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field and all data has been received for the set.
 7. Repeat steps 4–6 10 times.
 8. The Upper Tester disables advertising using the HCI_LE_Set_Extended_Advertising_Enable command for each set.
- Expected Outcome

Pass Verdict

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

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

For each set, the Lower Tester receives all the data sent in step 2, and no further data, in the AUX_ADV_IND PDUs and zero or more AUX_CHAIN_IND PDUs for each set.

4.2.2.34 LL/DDI/ADV/BV-36-C [AoD Connectionless CTE Advertising]

- Test Purpose

Tests that an advertiser IUT can send advertising packets with the AoD Connectionless Constant Tone Extension included with correct contents and with the correct event timing when utilizing a public device address. Advertisements without data, along with chaining, are tested.

The Lower Tester is configured to scan for an AoD Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension data. The Lower Tester observes the event timing and packet contents on the advertising channels in use.

- Reference

[\[13\] 2.5.2, 2.5.3](#)

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.

The maximum number of packets with CTE to be transmitted in a periodic advertising event is defined by the TSPX_max_CTE_count IXIT entry and the corresponding periodic advertising interval is defined by the TSPX_per_adv_interval IXIT entry.



- Test Procedure

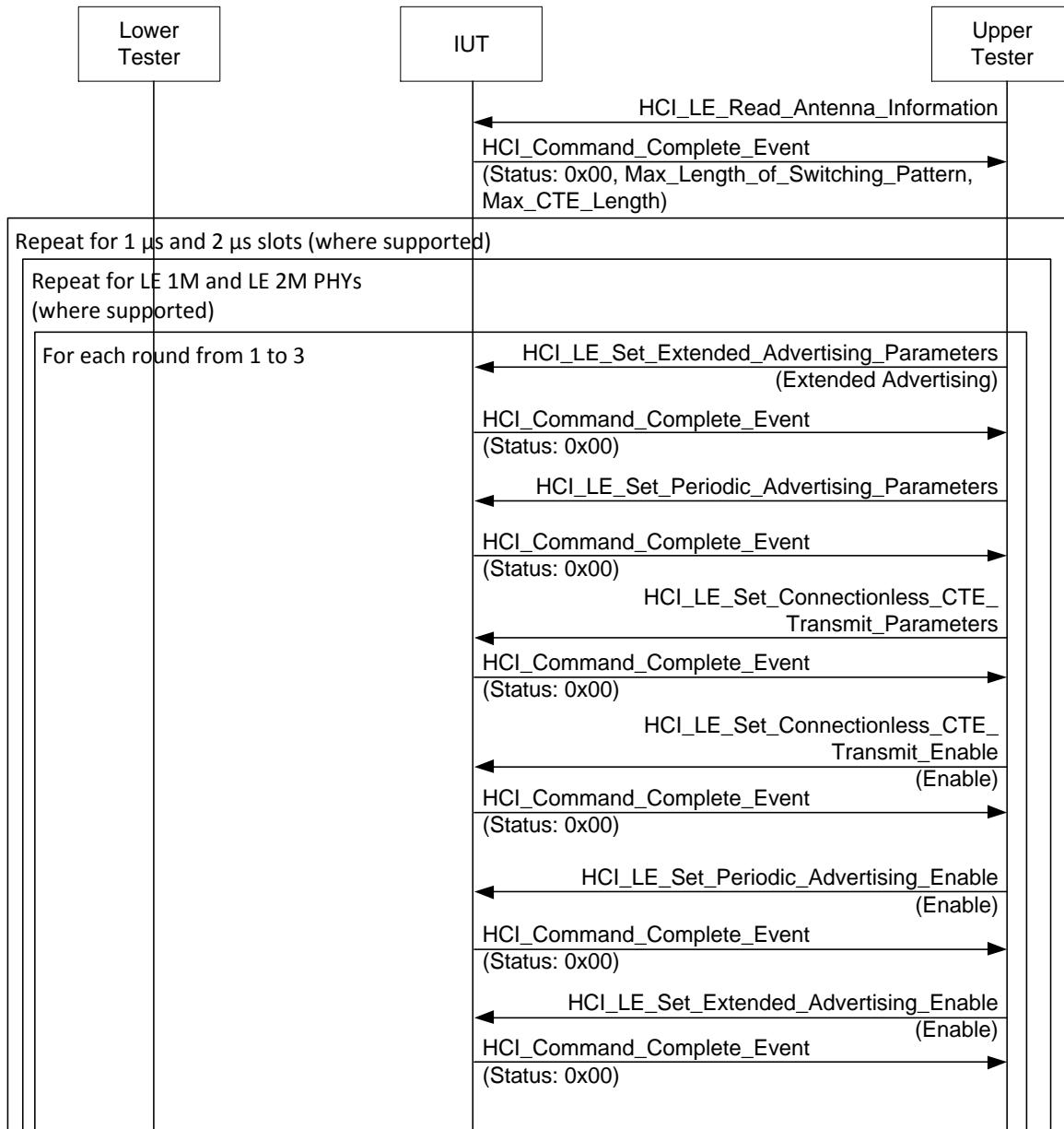


Figure 4.67: LL/DDI/ADV/BV-36-C [AoD Connectionless CTE Advertising] – Part A

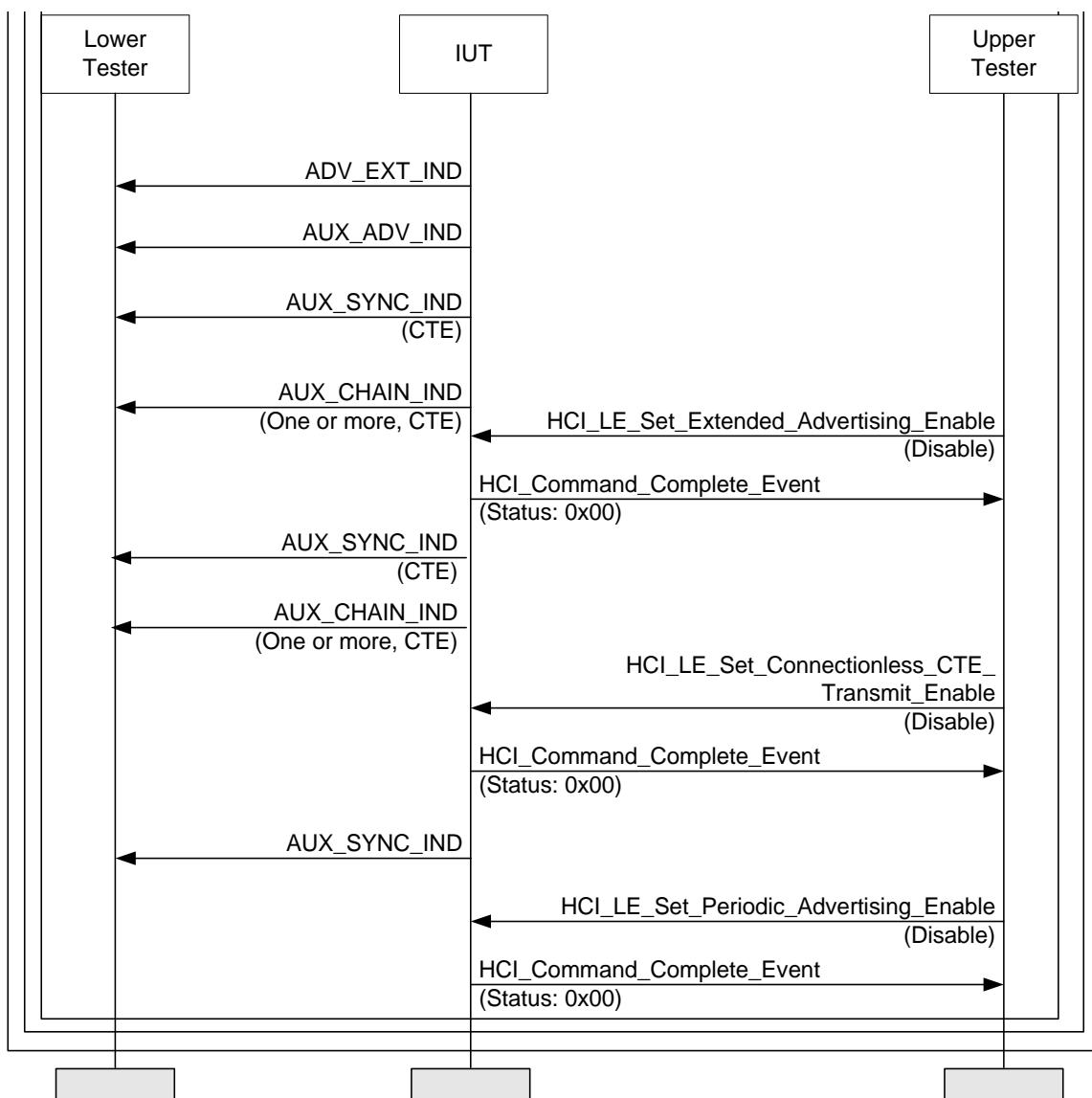


Figure 4.68: LL/DDI/ADV/BV-36-C [AoD Connectionless CTE Advertising] – Part B

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_Length_Switching_Pattern between 0x02 and 0x4B and a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_Length_Switching_Pattern and the Max_CTE_Length for future use.

For each round from 1 to 3 based on [Table 4.17](#):

Round	CTETime (Step 5)
1	0x02
2	0x0A
3	Max_CTE_Length from step 1

Table 4.17: Parameter values for each case variation

2. If the CTETime listed in [Table 4.17](#) for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 17.



3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and periodic advertising interval set to TSPX_per_adv_interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.
5. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 4. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots). CTE_Length shall be set to the value specified in [Table 4.17](#). CTE_Count shall be set to TSPX_max_CTE_count.
6. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).
7. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).
8. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).
9. The Lower Tester expects an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.
10. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.
11. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 μ s slots). The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5.
12. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 μ s slots). The packet containing the AUX_CHAIN_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field.
13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. The Lower Tester confirms that periodic advertising continues with CTE Transmission when extended advertising is disabled by repeating steps 11 and 12.
15. The Upper Tester disables CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command.



16. The Lower Tester confirms that periodic advertising continues without CTE Transmission when CTE Transmission is disabled by repeating steps 11 and 12, except that the packet containing the AUX_SYNC_IND PDU shall not contain the Constant Tone Extension field and the AUX_SYNC_IND PDU shall not contain the CTEInfo field or the AuxPtr field.
 17. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
 18. Repeat steps 2-17 for each round shown in [Table 4.17](#).
 19. If the IUT supports the LE 2M PHY, repeat steps 2-18, except that in step 3 the Primary_Advertising_PHY shall be set to 0x01 (LE 1M) and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).
 20. If the IUT supports 1 μ s slots, repeat steps 1-19, except that in step 5 the CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 μ s slots) and that in steps 11 and 12 the CTEType shall be set to 1 (AoD Constant Tone Extension with 1 μ s slots).
- Expected Outcome

Pass Verdict

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

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- The AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND advertisements.
- When CTE Transmission is enabled, the packets containing AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs and AUX_CHAIN_IND PDUs include the CTEInfo field.
- When CTE Transmission is disabled, the packets containing AUX_SYNC_IND PDUs do not include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs do not include the CTEInfo field or the AuxPtr field.
- The IUT sends a number of CTE_Count packets containing a Constant Tone Extension in each periodic advertising event.

[4.2.2.35 LL/DDI/ADV/BV-37-C \[AoA Connectionless CTE Advertising\]](#)

- Test Purpose

Tests that an advertiser IUT can send advertising packets with the AoA Connectionless Constant Tone Extension included with correct contents and with the correct event timing when utilizing a public device address. Advertisements without data, along with chaining, are tested.

The Lower Tester is configured to scan for an AoA Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension. The Lower Tester observes the event timing and packet contents on the advertising channels in use.

- Reference
- [\[13\] 2.5.2, 2.5.3](#)
- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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



The maximum number of packets with CTE to be transmitted in a periodic advertising event is defined by the TSPX_max_CTE_count IXIT entry and the corresponding periodic advertising interval is defined by the TSPX_per_adv_interval IXIT entry.

- Test Procedure

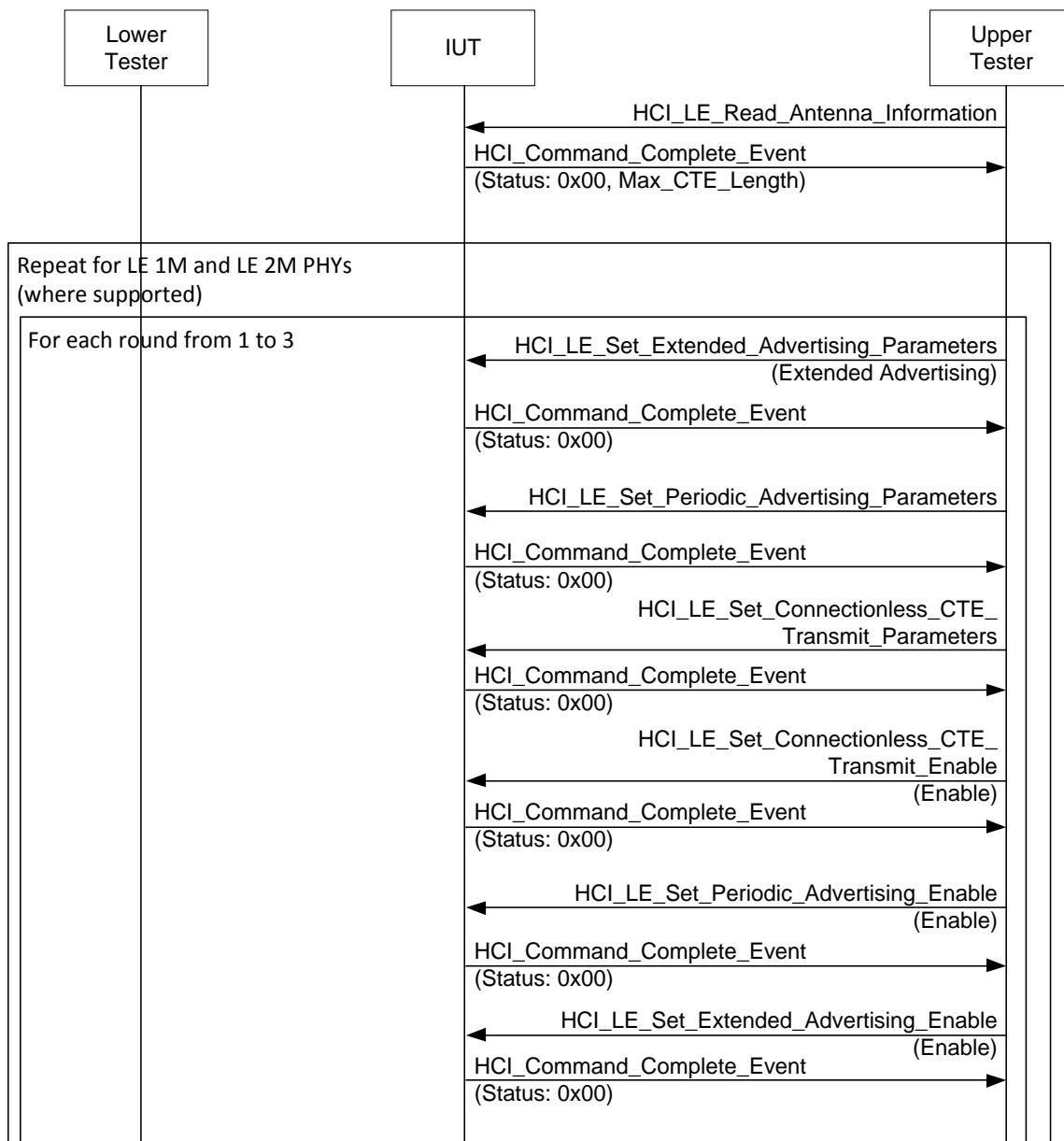


Figure 4.69: LL/DDI/ADV/BV-37-C [AoA Connectionless CTE Advertising] – Part A



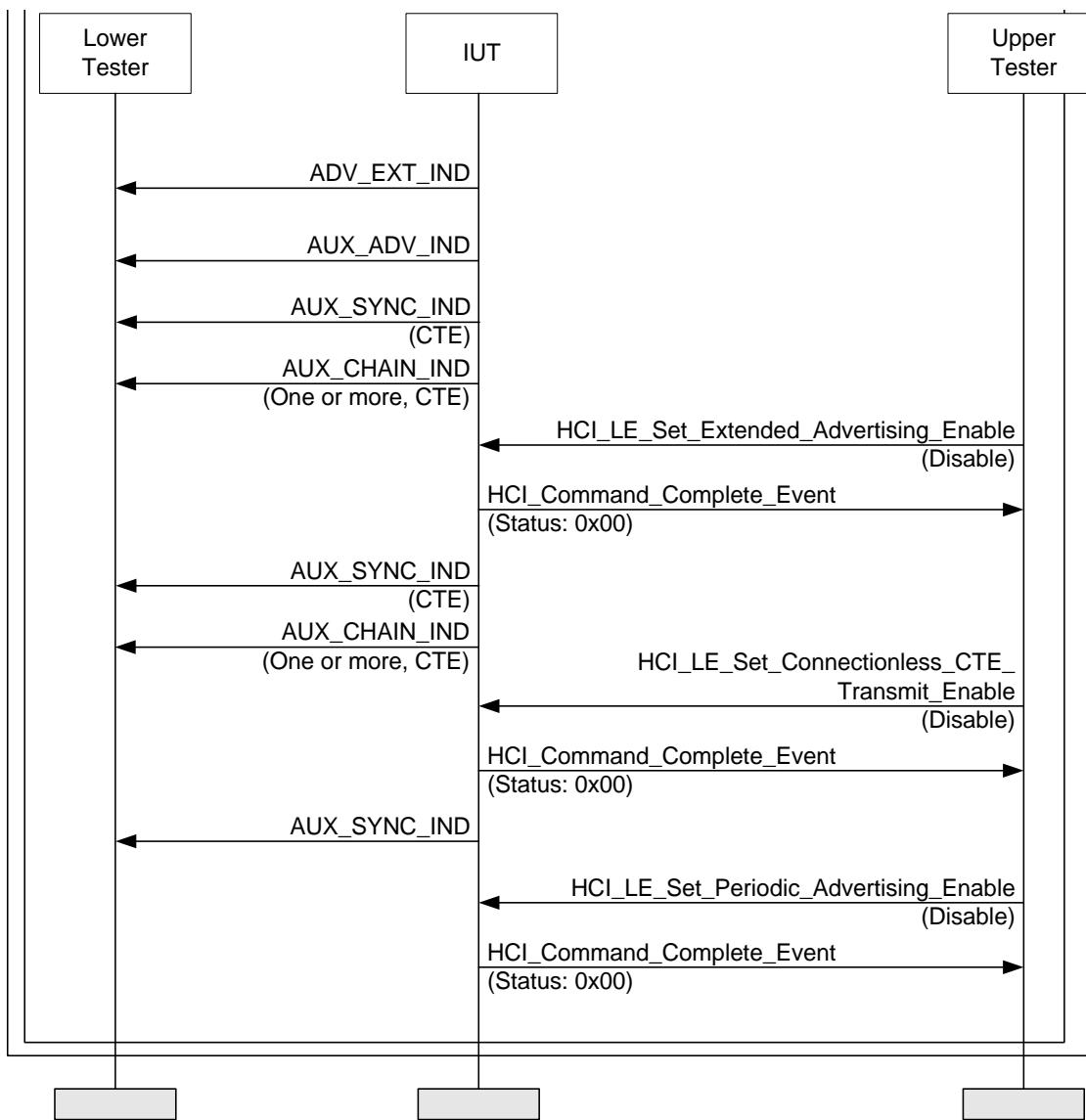


Figure 4.70: LL/DDI/ADV/BV-37-C [AoA Connectionless CTE Advertising] – Part B

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_CTE_Length for future use.

For each round from 1 to 3 based on [Table 4.18](#):

Round	CTETime (Step 5)
1	0x02
2	0x0A
3	Max_CTE_Length from step 1

Table 4.18: Parameter values for each case variation

2. If the CTETime listed in [Table 4.18](#) for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 17.



3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and periodic advertising interval set to TSPX_per_adv_interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.
5. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 4. CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). CTE_Length shall be set to the value specified in [Table 4.18](#). CTE_Count shall be set to TSPX_max_CTE_count.
6. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).
7. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).
8. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).
9. The Lower Tester expects an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.
10. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.
11. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5.
12. If the AUX_SYNC_IND PDU contains an AuxPtr field, the Lower Tester utilizes it to listen for an AUX_CHAIN_IND PDU with the AdvMode field set to 00b. The AUX_CHAIN_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 5, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 μ s slots). The packet containing the AUX_CHAIN_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 5. If the AUX_CHAIN_IND PDU contains an AuxPtr field this step is repeated until an AUX_CHAIN_IND PDU is received with no AuxPtr field.
13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. The Lower Tester confirms that periodic advertising continues with CTE Transmission when extended advertising is disabled by repeating steps 11 and 12.
15. The Upper Tester disables CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command.
16. The Lower Tester confirms that periodic advertising continues without CTE Transmission when CTE Transmission is disabled by repeating steps 11 and 12, except that the packet containing the AUX_SYNC_IND PDU shall not contain the Constant Tone Extension field and the AUX_SYNC_IND PDU shall not contain the CTEInfo field or the AuxPtr field.



17. The Upper Tester disables periodic advertising using the `HCI_LE_Set_Periodic_Advertising_Enable` command.
 18. Repeat steps 2-17 for each round shown in [Table 4.18](#).
 19. If the IUT supports the LE 2M PHY, repeat steps 2-18, except that in step 3 the `Primary_Advertising_PHY` shall be set to 0x01 (LE 1M) and the `Secondary_Advertising_PHY` shall be set to 0x02 (LE 2M).
- Expected Outcome

Pass Verdict

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

- The IUT sends an `ADV_EXT_IND` with an `AuxPtr` field referring to an `AUX_ADV_IND` on the secondary advertising channel.
- The `AUX_ADV_IND` includes a `SynInfo` field containing synchronization information for the periodic `AUX_SYNC_IND` advertisements.
- When CTE Transmission is enabled, the packets containing `AUX_SYNC_IND` PDUs or `AUX_CHAIN_IND` PDUs include the Constant Tone Extension field, and the `AUX_SYNC_IND` PDUs and `AUX_CHAIN_IND` PDUs include the `CTEInfo` field.
- When CTE Transmission is disabled, the packets containing `AUX_SYNC_IND` PDUs do not include the Constant Tone Extension field, and the `AUX_SYNC_IND` PDUs do not include the `CTEInfo` field or the `AuxPtr` field.
- The IUT sends a number of `CTE_Count` packets containing a Constant Tone Extension in each periodic advertising event.

[4.2.2.36 LL/DDI/ADV/BV-39-C \[Connectionless CTE Advertising – Maintain CTE Configuration\]](#)

- Test Purpose

Tests that an advertiser IUT configured to send advertising packets with Constant Tone Extension and with disabled periodic advertisements starts sending Constant Tone Extensions after periodic advertisements are re-enabled.

The Lower Tester is configured to scan for Connectionless Constant Tone Extension. The Upper Tester configures the IUT to generate a Connectionless Constant Tone Extension.

- Reference

[\[13\]](#) 2.5.2, 2.5.3

- Initial Condition

Parameters: `LL_advertiser_advInterval_MIN`, `LL_advertiser_advInterval_MAX`, `LL_advertiser_Adv_Channel_Map`

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



- Test Procedure

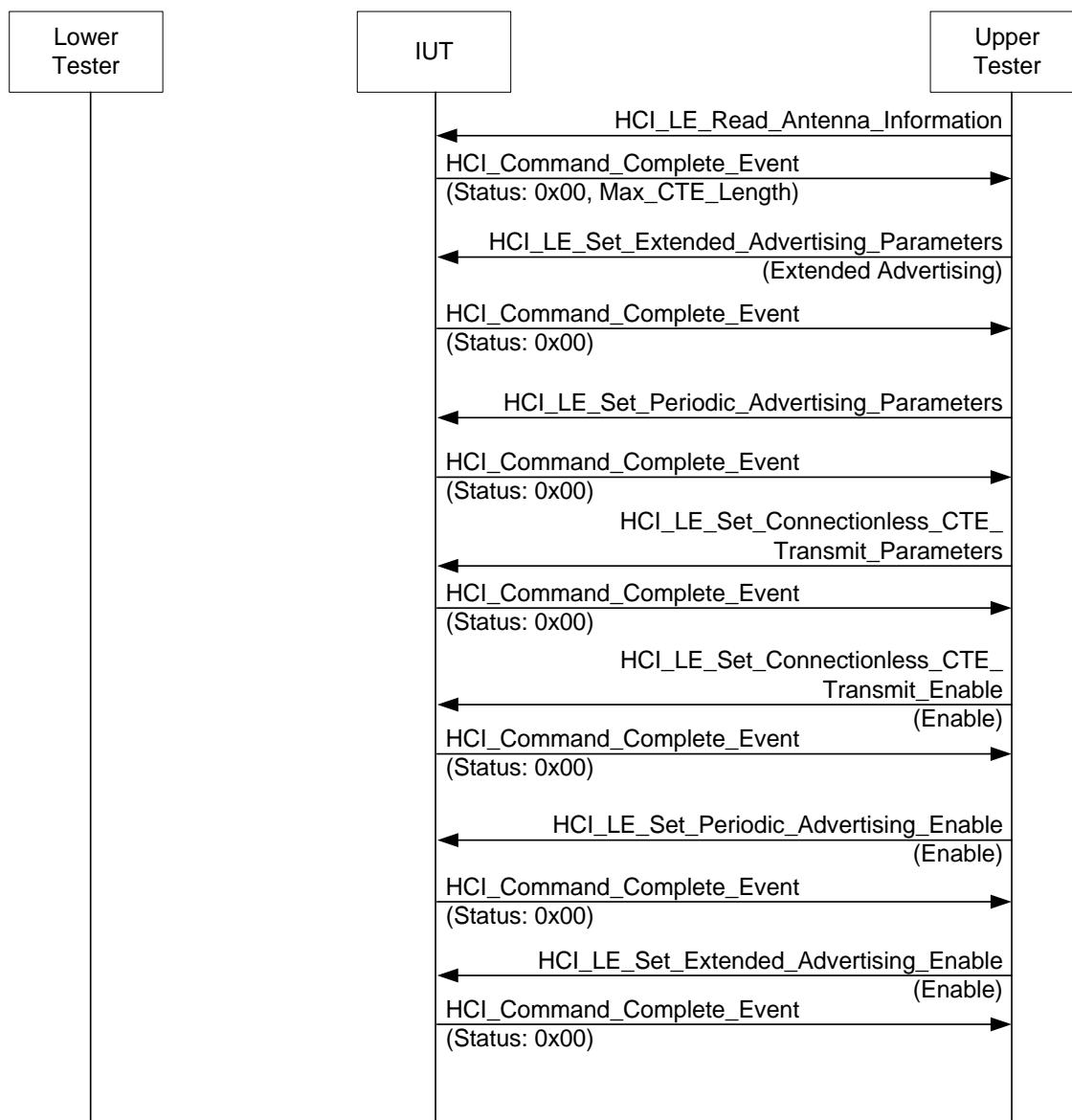


Figure 4.71: LL/DDI/ADV/BV-39-C [Connectionless CTE Advertising - Maintain CTE Configuration] – Part A

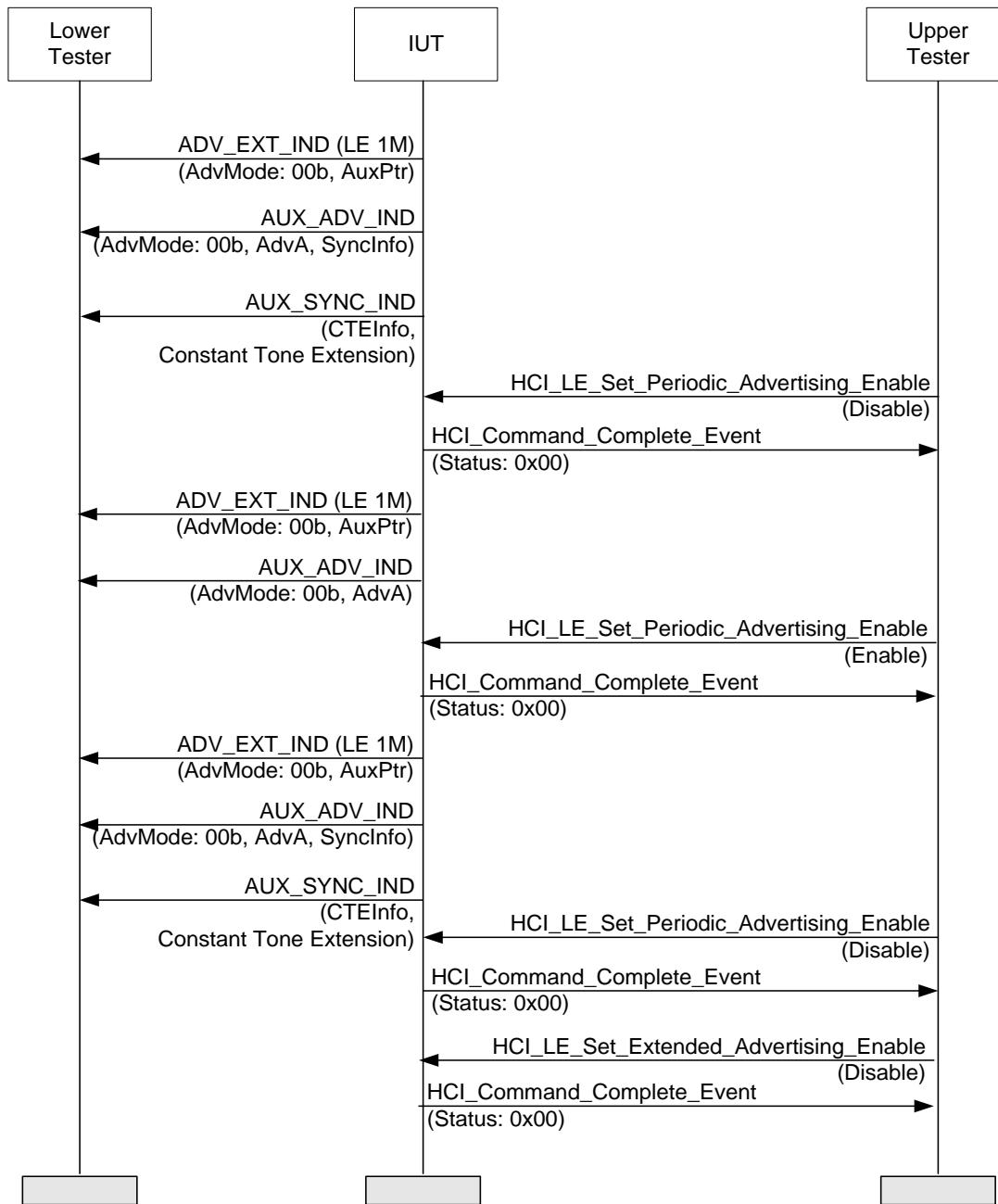


Figure 4.72: LL/DDI/ADV/BV-39-C [Connectionless CTE Advertising - Maintain CTE Configuration] – Part B

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14.
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 0x0000. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).

3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Periodic_Advertising_Properties parameter shall be set to 0x0000.
 4. The Upper Tester sends an HCI_LE_Set_Connectionless_CTE_Transmit_Parameters command to the IUT. Advertising_Handle shall be set to the handle used in step 3. CTE_Type and CTE_Length shall be set to a valid value supported by the IUT.
 5. The Upper Tester enables Connectionless CTE Transmission using the HCI_LE_Set_Connectionless_CTE_Transmit_Enable command with the CTE_Enable parameter set to 0x01 (enabled).
 6. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).
 7. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration parameter shall be set to 0x0000 (No Advertising Duration).
 8. The Lower Tester expects an ADV_EXT_IND PDU from the IUT on the primary advertising channel. The ADV_EXT_IND PDU shall contain the AuxPtr field and shall not contain the CTEInfo field.
 9. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall contain the SyncInfo field and shall not contain the CTEInfo field.
 10. The Lower Tester utilizes the SyncInfo field to listen for a packet containing an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements. The AUX_SYNC_IND PDU shall contain the CTEInfo field, with CTETime set to the CTE_Length value from step 4, RFU set to '0', and the CTEType set to the CTE_Type value from step 4. The packet containing the AUX_SYNC_IND PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the time specified in CTE_Length from step 4.
 11. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
 12. The Lower Tester confirms that extended advertising continues by repeating step 8.
 13. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel. The AUX_ADV_IND PDU shall not contain the SyncInfo or the CTEInfo field.
 14. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command with the Enable parameter set to 0x01 (enabled).
 15. The Lower Tester confirms that extended advertising continues by repeating steps 8-10.
 16. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
 17. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
- Expected Outcome

Pass Verdict

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

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- When periodic advertisement is enabled, the AUX_ADV_IND includes a SyncInfo field containing synchronization information for the periodic AUX_SYNC_IND.
- When periodic advertisement is disabled, the AUX_ADV_IND does not include a SyncInfo field.



- The packets containing AUX_SYNC_IND PDUs include the Constant Tone Extension field, and the AUX_SYNC_IND PDUs include the CTEInfo field.

4.2.2.37 LL/DDI/ADV/BV-43-C [Periodic Advertising validating SyncInfo fields]

- Test Purpose

Tests that the AA and CRCInit fields shall be the same for all SyncInfo fields describing the same periodic advertising. Non-Connectable and Non-Scannable advertisements are tested. The Lower Tester confirms that the AA and CRCInit fields in all SyncInfo fields for the same periodic advertising train are identical.

- Reference

[10] 4.2.2.24

- Initial Condition

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

- Test Procedure

Execute the test procedure using non-connectable and non-scannable periodic advertising events with a selected periodic advertising interval. Confirm that the AUX_ADV_IND packets received by the Lower Tester have identical AA and CRCInit fields in the SyncInfo parameter.



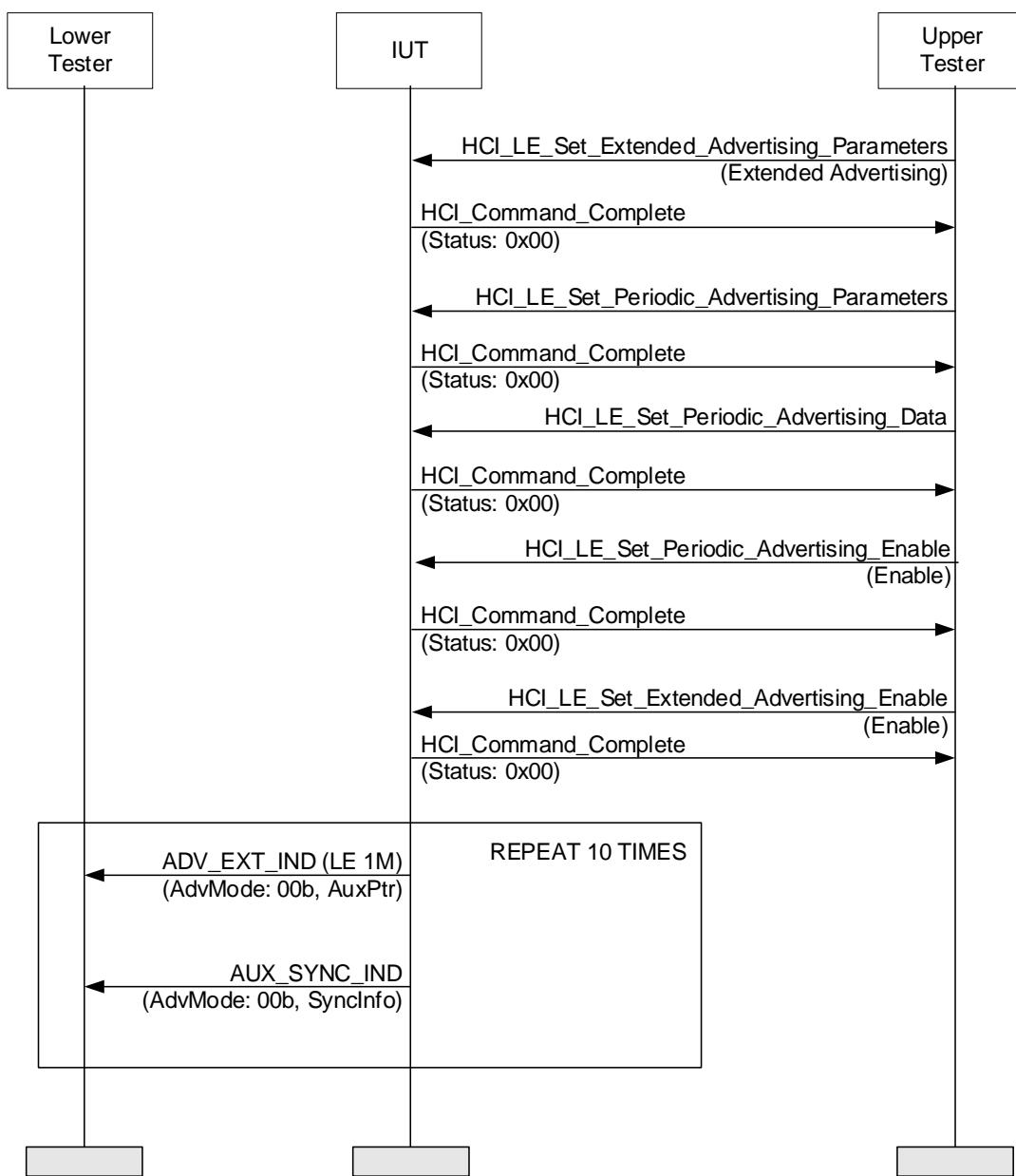


Figure 4.73: LL/DDI/ADV/BV-43-C [Periodic Advertising validating SyncInfo fields]

1. The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command to the IUT using all supported advertising channels. `Advertising_Event_Properties` parameter shall be set to 0x0000. The `Own_Address_Type` shall be set to 0x00 (Public Device Address). The `Primary_Advertising_PHY` shall be set to 0x01 (LE 1M). The `Secondary_Advertising_PHY` shall be set to 0x01 (LE 1M).
1. The Upper Tester sends an `HCI_LE_Set_Periodic_Advertising_Parameters` command to the IUT using minimum advertising interval. `Periodic_Advertising_Properties` parameter is set to 0x0000.
2. The Upper Tester sends one `HCI_LE_Set_Periodic_Advertising_Data` command to the IUT with a data length of 0x01 and value of 0x00.
3. The Upper Tester enables periodic advertising using the `HCI_LE_Set_Periodic_Advertising_Enable` command with the `Enable` parameter set to 0x01 (Periodic Advertising).
4. The Upper Tester enables advertising using the `HCI_LE_Set_Extended_Advertising_Enable` command. The `Duration[0]` parameter is set to 0x0000 (No Advertising Duration).

5. The Lower Tester expects an ADV_EXT_IND packet from the IUT with AdvMode set to 00b with the AuxPtr Extended Header field present.
 6. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the AdvMode field set to 00b and the SyncInfo Extended Header fields present. Note: There can be AUX_SYNC_IND PDUs transmitted forming the periodic advertising.
 7. Repeat steps 6 and 7 10 times.
- Expected Outcome

Pass Verdict

- The Lower Tester confirms the AA and CRCInit fields in the SyncInfo extended payload are identical for each SyncInfo field in each AUX_ADV_IND PDU.

4.2.2.38 LL/DDI/ADV/BI-05-C [Disallow Extended Advertising PDU sizes for Legacy Advertising when advertising enabled]

- Test Purpose

Verify that a supported legacy command yields a Command Complete event with status 'Invalid HCI Command Parameters' in return when sending PDU sizes larger than legacy advertising PDU after advertising has been enabled.

- Reference

[1] 7.8.54, 7.8.56

- Initial Condition

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

- Test Procedure

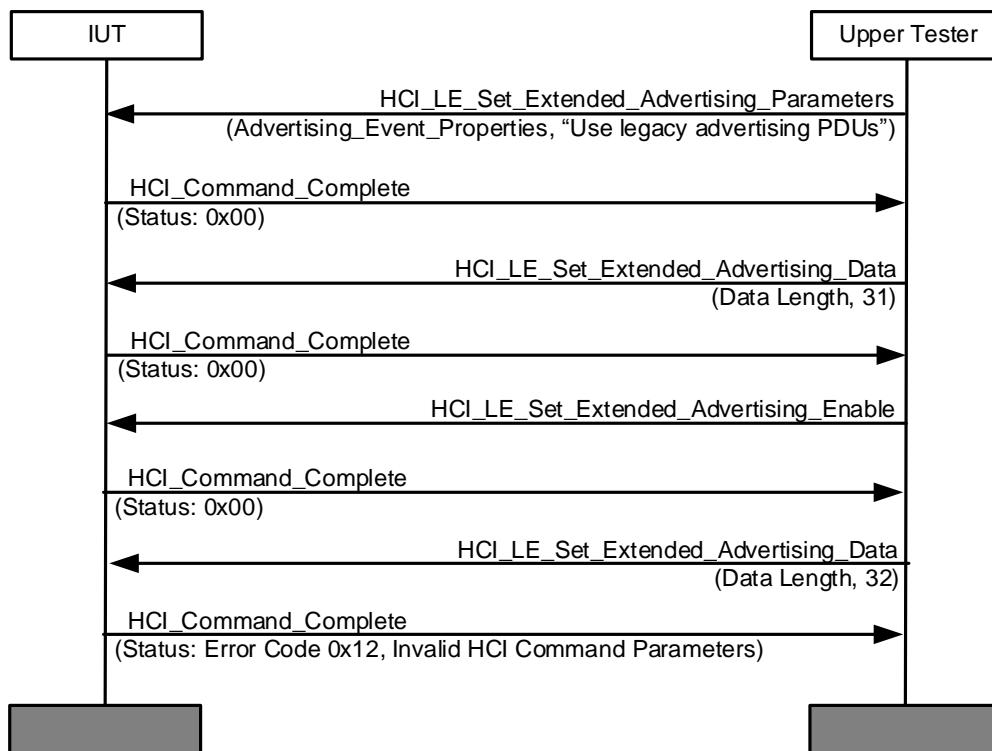


Figure 4.74: LL/DDI/ADV/BI-05-C [Disallow Extended Advertising PDU sizes for Legacy Advertising when advertising enabled]



1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command with Advertising_Event_Properties set to “Use legacy advertising PDUs”.
 2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command with Data Length equal to 31 to the IUT. The IUT sends a Command Complete event with Status set to 0x00.
 3. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable to the IUT. The IUT is expected to return an HCI_Command_Complete event with Error Code 0x00 (Success).
 4. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Data command with Data Length equal to 32 to the IUT. The IUT sends a Command Complete event with Status set to 0x12.
- Expected Outcome

Pass Verdict

The IUT returns an HCI Command Complete event with Status = Invalid HCI Command Parameters after the HCI_LE_Set_Extended_Advertising_Data command in step 4 is received.

4.2.2.39 LL/DDI/ADV/BI-06-C [Disallow Extended Advertising PDU sizes for Scannable Legacy Advertising when advertising enabled]

- Test Purpose

Verify that the HCI_LE_Set_Extended_Scan_Response_Data command yields a Command Complete event with status ‘Invalid HCI Command Parameters’ in return when sending scannable legacy PDU sizes larger than legacy advertising PDU after advertising has been enabled.

- Reference

[1] 7.8.55, 7.8.56

- Initial Condition

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



- Test Procedure

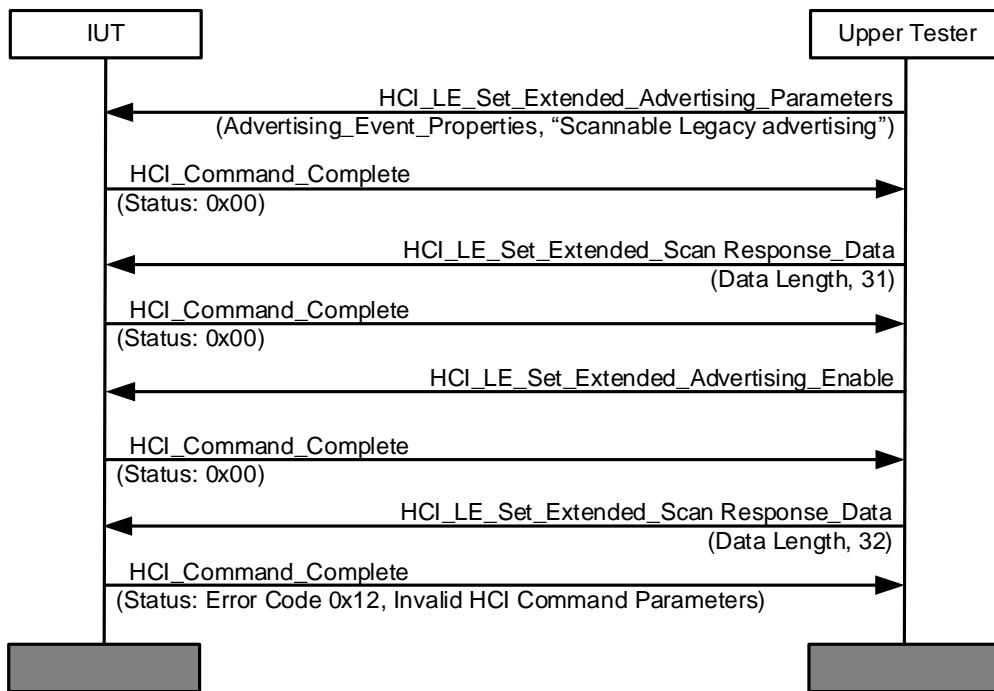


Figure 4.75: LL/DDI/ADV/BI-06-C [Disallow Extended PDU sizes for Scannable Legacy Advertising when advertising enabled]

1. The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command with `Advertising_Event_Properties` set to “Scannable Legacy advertising” and “Use legacy advertising PDUs”.
2. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Response_Data` command with Data Length equal to 31 to the IUT. The IUT sends a Command Complete event with Status set to 0x00.
3. The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Enable` to the IUT. The IUT is expected to return an `HCI_Command_Complete` event with Error Code 0x00 (Success).
4. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Response_Data` command with Data Length equal to 32 to the IUT. The IUT sends a Command Complete event with Status set to 0x12.

- Expected Outcome

Pass Verdict

The IUT returns an `HCI_Command_Complete` event with Status = Invalid HCI Command Parameters after the `HCI_LE_Set_Scan_Response_Data` command in step 4 is received.

4.2.3 SCN

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

4.2.3.1 Common PDU Contents

The default advertising channel packet contents for the Lower Tester sent and Upper Tester accepted packets are defined in ‘/LL/DDI’.

The addresses used vary per test and the data is typically a count of the events transmitted by the Lower Tester. The Lower Tester uses the device name “LT” in scan response packets.



4.2.3.2 LL/DDI/SCN/BV-01-C [Passive Scanning: Non-connectable]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using non-connectable advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester. The advertising packets' payload is a sequence numbering of the packets.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.



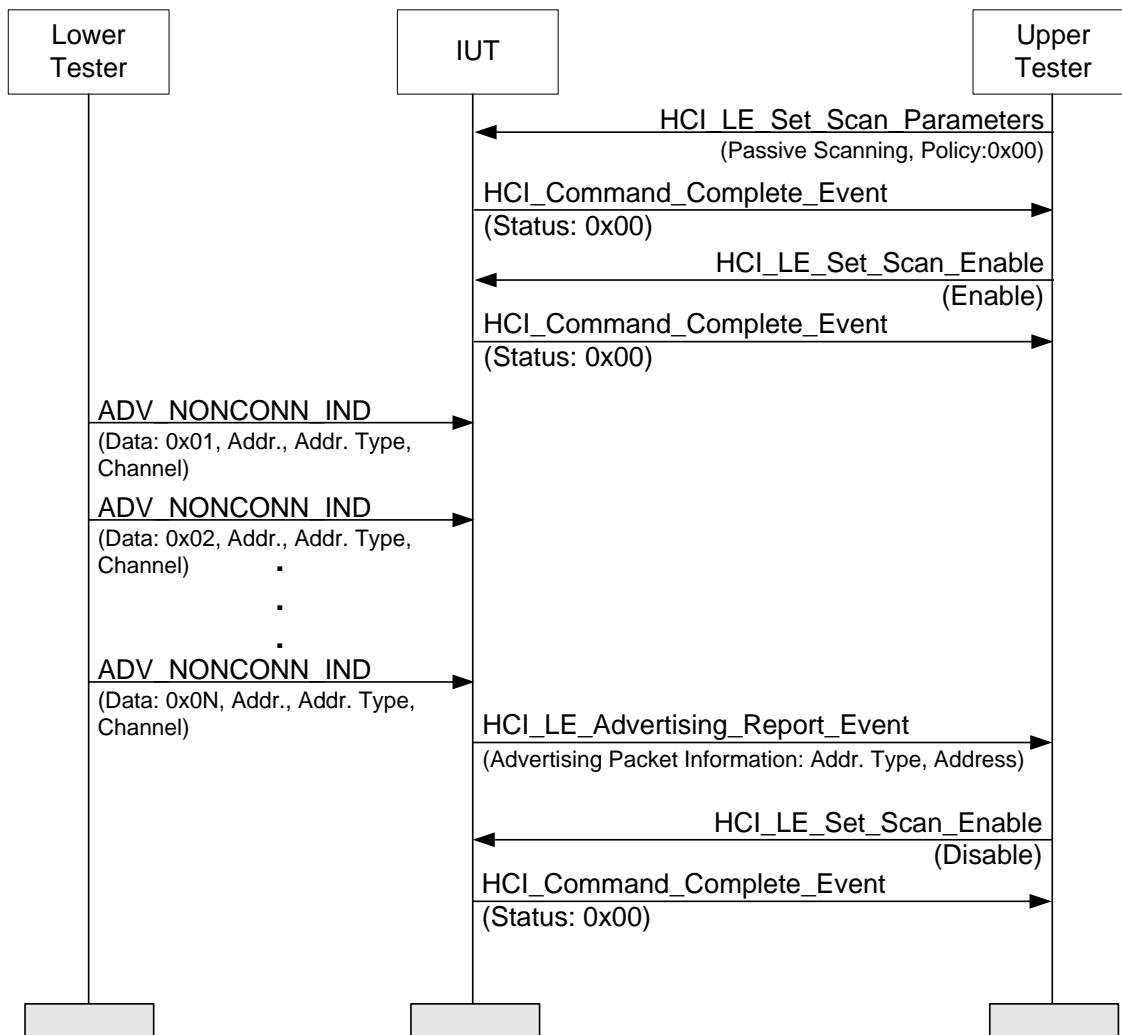


Figure 4.76: LL/DDI/SCN/BV-01-C [Passive Scanning: Non-connectable]

1. Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
2. Upper Tester enables passive scanning in the IUT.
3. Lower Tester sends ADV_NONCONN_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
4. Upper Tester expects an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in Step 3 and data included in one of the advertising packets.
5. Upper Tester sends an HCI LE Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
6. Configure Lower Tester as advertiser using the second supported advertising channel and an address with an identical company identifier but a different company assigned identifier than the IUT
7. Repeat steps 2-5.
8. Configure Lower Tester as advertiser using the third supported advertising channel using an address with a different company identifier and using the same company assigned identifier as the IUT and a random address type.
9. Repeat steps 2-5.

- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel separately, with the IUT recognizing each advertising packet type.

4.2.3.3 LL/DDI/SCN/BV-02-C [Passive Scanning Device Filtering]

- Test Purpose

Tests that a scanner IUT is able to scan for a specific advertiser using device filtering.

The Lower Tester advertises using non-connectable events on one channel and expects the IUT to report the information in the advertising packets. The advertising packets' payload is a sequence numbering of the packets sent.

- Reference

[3] 4.3.3, 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX.

State: Passive Scanning (selected scan interval, selected scan window) AND Specific White Listed
(one white listed device address, public address type, policy for scanner, black list all unknown devices)

- Test Procedure

Execute the test procedure using the minimum advertising interval and the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval. Use the filtering policy to black list all unknown devices (Ignore advertising packets from devices not in the White List Only (0x01)).



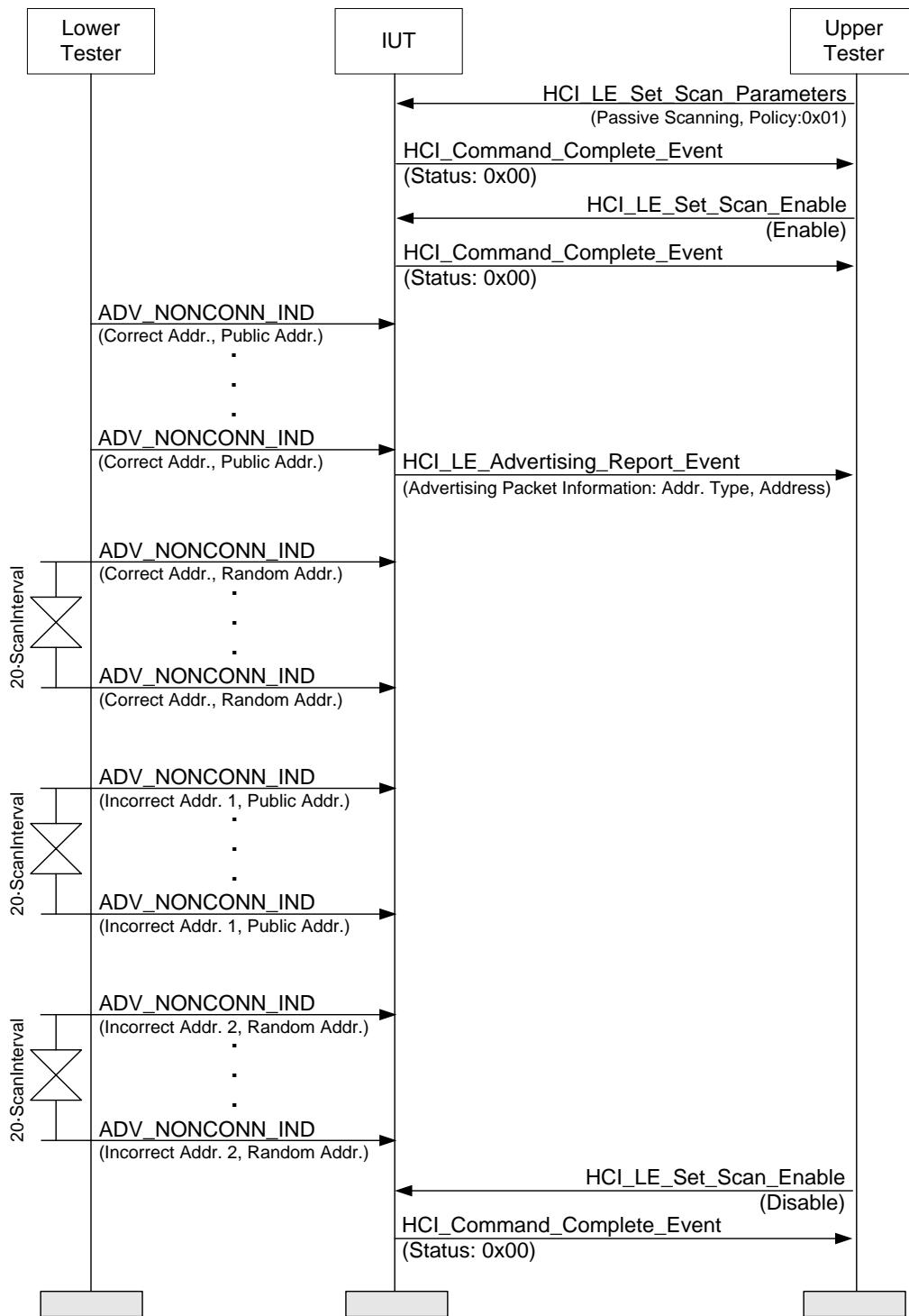


Figure 4.77: LL/DDI/SCN/BV-02-C [Passive Scanning Device Filtering]

1. Upper Tester enables passive scanning in the IUT with the filtering policy set to 'Ignore advertising packets from devices not in the White List Only (0x01)'.
2. Configure Lower Tester as advertiser, using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
3. Lower Tester sends ADV_NONCONN_IND packets each advertising event on the selected advertising channel only. Repeat until 20 scan intervals have passed or Step 4 executes.

4. Upper Tester expects an HCI_LE_Advertising_Report event from the IUT which indicates use of non-connectable advertising events, a public address type, and the white listed device address.
 5. Disable advertising in Lower Tester.
 6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a random address type
 7. Lower Tester sends ADV_NONCONN_IND packets each advertising event on the selected advertising channel only. Repeat until 20 scan intervals have passed.
 8. Disable advertising in Lower Tester.
 9. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type. The device address has a different company assigned identifier and the same company identifier as IUT.
 10. Repeat Steps 7-8.
 11. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a random address type. The device address has a different company assigned identifier and company identifier as IUT.
 12. Repeat Steps 7-8.
 13. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and expect an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The test procedure completes using a specific advertising channel with the timing combination.

The IUT reports the device address matching the white listed both for the address and type with an HCI event,

The IUT does not report the device address matching the white listed for the address and but mismatching the type with an HCI event,

The IUT does not report the device addresses mismatching the white listed address with an HCI event.

4.2.3.4 LL/DDI/SCN/BV-03-C [Active Scanning]

- Test Purpose

Tests that a scanner IUT detects and requests additional information from advertisers and reports the results from the Controller.

The Lower Tester advertises using connectable undirected events on one channel at a time, acting as multiple advertisers, expects the IUT to send the request, then responds to the IUT. The advertising packets' payload is a sequence numbering of the packets sent. The Lower Tester observes the packets and timing of the IUT, as well as the HCI events reporting scan results.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))



- Test Procedure

Execute the test procedure advertising on a selected advertising channel at a time, using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

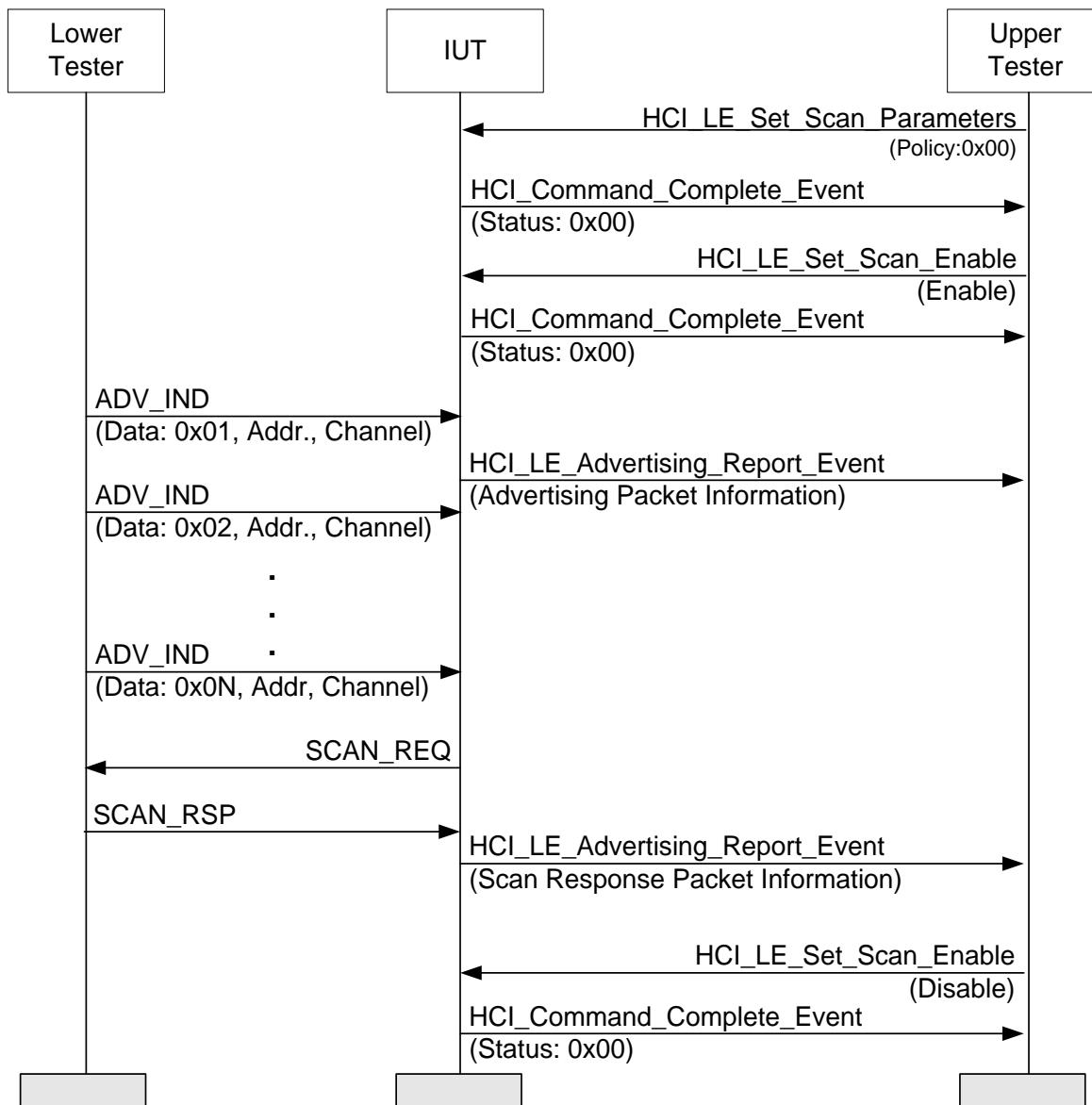


Figure 4.78: LL/DDI/SCN/BV-03-C [Active Scanning]

- Configure Lower Tester advertising channel map using the first supported advertising channel.
- Configure Lower Tester as advertiser using a common device address and a public address type.
- Upper Tester enables active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.
- Lower Tester sends ADV_IND packets with event count as data encoded least significant bit first, each advertising event on the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20) or Step 5 executes.
- Lower Tester expects an SCAN_REQ packet T_IFS after any of the ADV_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after of the SCAN_REQ packet.



6. Interleave with Step 4: Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the ADV_IND packets.
 7. Interleave with Step 5: Upper Tester expects an HCI_LE_Advertising_Report event containing the scan response information sent in Step 5.
 8. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 9. Configure Lower Tester as advertiser using a device address with an identical company assigned identifier but a different company identifier than IUT address and a public address type.
 10. Repeat steps 3-8.
 11. Configure Lower Tester as advertiser using a device address with an identical company identifier but a different company assigned identifier than IUT address and a public address type.
 12. Repeat steps 3-8.
 13. Configure Lower Tester as advertiser using a device address with different company identifier and company assigned identifier than IUT address and a public address type.
 14. Repeat steps 3-8.
 15. Configure Lower Tester advertising channel map using the second supported advertising channel.
 16. Repeat steps 2-14.
 17. Configure Lower Tester advertising channel map using the third supported advertising channel.
 18. Repeat steps 2-14.
- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel separately with the IUT sending a scan request to each advertiser device,

The IUT reports the advertisers with both HCI events,

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

- Notes

Note that neither the order nor composition with which the advertising packets are reported by the scanner is fixed.

The Controller may queue the advertising reports and some information during a scan in one HCI LE Advertising Report event.

The test procedure may be executed with filtering set to accept only the Lower Tester address or with the filtering set to white list all unknown devices. In the latter case test realization has to allow detection of other devices than the test configuration.

4.2.3.5 LL/DDI/SCN/BV-04-C [Active Scanning Device Filtering]

- Test Purpose

Tests that a scanner IUT detects, requests and reports additional information about a single advertiser according to the filtering policy and type of advertising event used.

The Lower Tester advertises using connectable undirected events on one channel at a time, expects the IUT to send the request, then when applicable responds to the IUT. The advertising packets' payload is a sequence numbering of the packets sent.

- Reference

[3] 4.3.3, 4.4.3.2



- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (public address, selected scan interval, selected scan window) AND Specific White Listed (one white listed device address, one public type address, policy for scanner, black list all unknown devices)

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.



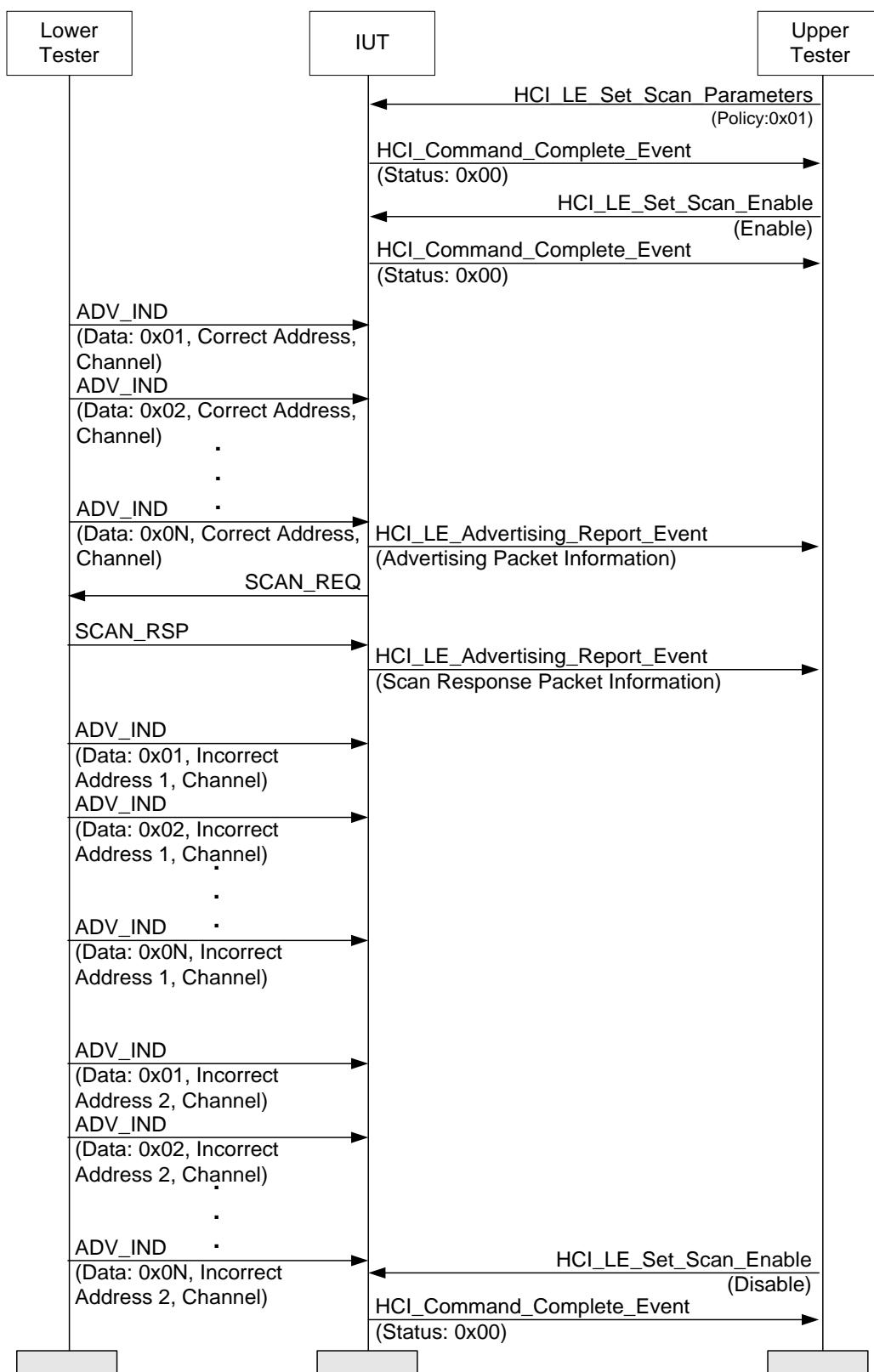


Figure 4.79: LL/DDI/SCN/BV-04-C [Active Scanning: Device Filtering]

1. Upper Tester enables active scanning with filtering policy set to 'Ignore advertising packets from devices not in the White List Only (0x01)' in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel.

3. Configure Lower Tester as advertiser using the white listed device address and a public address type. Scan response is also configured in the Lower Tester.
 4. Lower Tester sends ADV_IND packets each advertising event using the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20) or until Step 4 executes.
 5. Lower Tester expects an SCAN_REQ packet TIFS after any of the ADV_IND packets and sends an SCAN_RSP packet to the IUT TIFS after the SCAN_REQ.
 6. Interleave with Step 3: Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the advertising packets.
 7. Interleave with Step 4: Upper Tester expects an HCI_LE_Advertising_Report containing the response sent in Step 4.
 8. Configure Lower Tester as advertiser using a device address other than the white listed device address and a public address type. The device address has a different company assigned identifier and the same company identifier as IUT. Scan response is also configured in the Lower Tester.
 9. Lower Tester sends ADV_IND packets each advertising event using the selected advertising channel only, using the selected advertising interval. Repeat up to a number of scan intervals (20).
 10. Expect no SCAN_REQ in response after TIFS to any of the packets.
 11. Configure Lower Tester as advertiser using a device address other than the white listed device address and a public address type. The device address has a different company assigned identifier and company identifier as IUT. Scan response is also configured in the Lower Tester.
 12. Repeat steps 8-9
 13. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and expect an HCI_Command_Complete event in response.
 14. Configure Lower Tester advertising channel map using the second supported advertising channel.
 15. Repeat steps 2-13
 16. Configure Lower Tester advertising channel map using the third supported advertising channel.
 17. Repeat steps 2-13.
- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel with the timing combination, with the IUT sending a request to the white listed devices,

The IUT reports the information in the advertising packets of the white listed devices

4.2.3.6 LL/DDI/SCN/BV-05-C [Scanning For Advertiser Types]

- Test Purpose

Tests that a scanner IUT detects devices using different types advertising events when scanning actively and not filtering devices.

The Lower Tester sends advertising packets using several device addresses on multiple channels, using different responses and types of advertising events and observes the IUT reporting the advertisers. Some of the advertising packets include data.

- Reference

[3] 4.4.3.1, 4.4.3.2



- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))

- Test Procedure

Execute the test advertising using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).



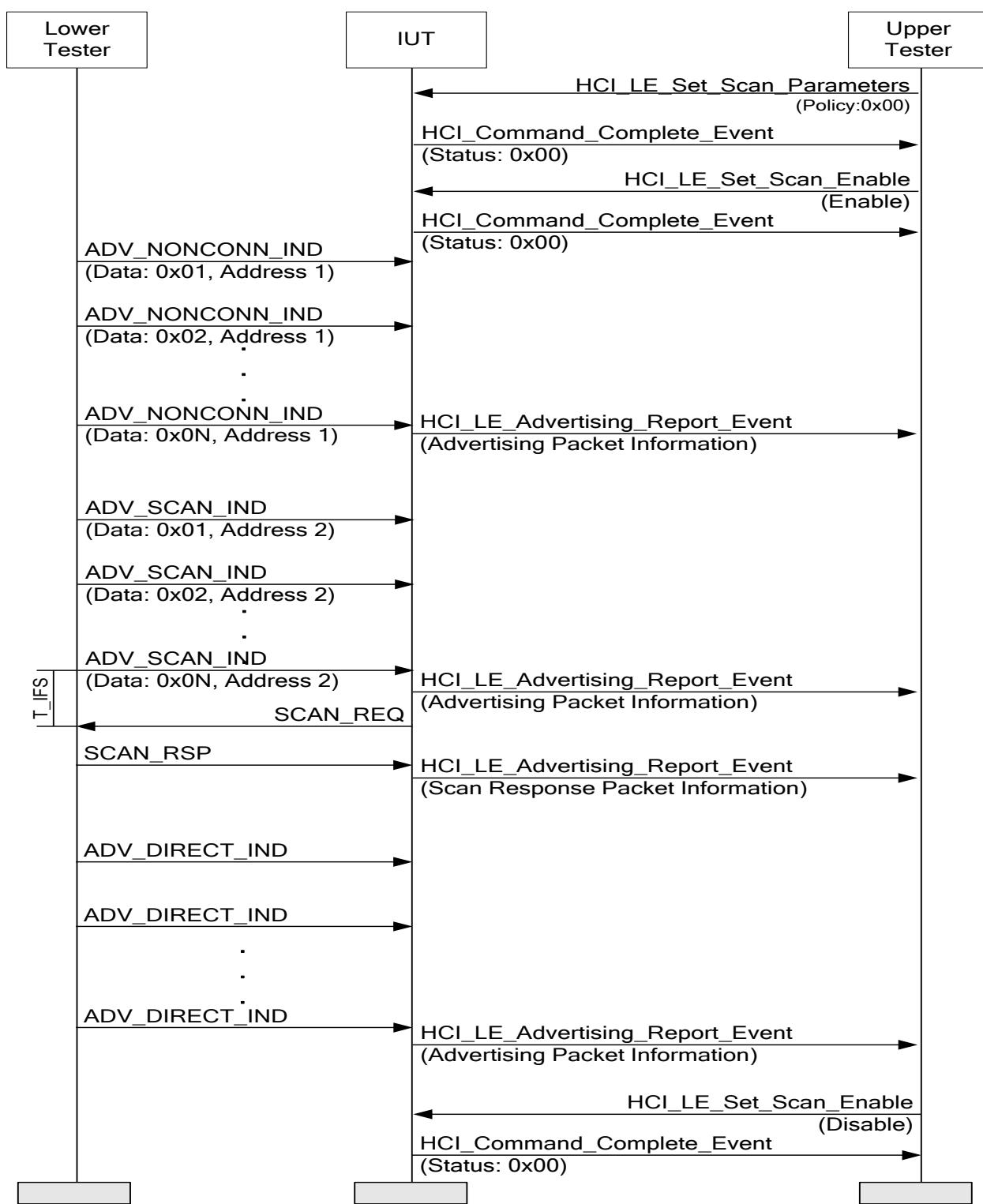


Figure 4.80: LL/DDI/SCN/BV-05-C [Scanning for Advertising types]

1. Upper Tester enables active scanning in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel, a device address with an identical company assigned identifier but different company identifier than the IUT and non-connectable advertising packet type.

3. Lower Tester sends ADV_NONCONN_IND packets each advertising event. Repeat up to a number of scan intervals (20) or Step 4 is executed.
 4. Upper Tester expects at least one HCI_LE_Advertising_Report Event containing the information used in the ADV_NONCONN_IND packets.
 5. Configure Lower Tester advertising channel map using the second supported advertising channel, a device address with a different company assigned identifier and the same company identifier as the IUT and undirected advertising packet type.
 6. Lower Tester sends ADV_SCAN_IND packets in each advertising event using the selected advertising interval. Repeat until Step 7 or up to a number of scan intervals (30).
 7. Lower Tester expects an SCAN_REQ packet TIFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet to the IUT TIFS after the SCAN_REQ.
 8. Interleave with Step 6: Upper Tester expects an HCI_LE_Advertising_Report Event containing the advertising packet information.
 9. Interleave with Step 7: Upper Tester expects an HCI_LE_Advertising_Report Event containing the Lower Tester address in the response sent in Step 7.
 10. Configure Lower Tester advertising channel map using the third supported advertising channel, a device address with a different company assigned identifier and company identifier as the IUT and directed advertising packet type
 11. Lower Tester sends ADV_DIRECT_IND packets with the IUT device address for the initiator, each advertising event using the directed advertising packet interval. Repeat up to a number of scan intervals (15) or Step 8 is executed.
 12. Upper Tester expects an HCI_LE_Advertising_Report Event containing the advertising packet information.
 13. Upper Tester sends an HCI_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The test procedure completes applying each advertiser address and advertising event type.

The IUT reports the packet information for the non-connectable and directed and undirected connectable packets to which a response packet was not sent.

The IUT reports the connectable packets to which a response packet was sent.

- Notes

The test procedure may be executed with filtering set to accept only the Lower Tester address or with the filtering set to white list all unknown devices. In the latter case test realization has to allow detection of other devices than the test configuration.

4.2.3.7 LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected Events]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using undirected advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester. The advertising packets' payload is a sequence numbering of the packets.

- Reference

[3] 4.4.3.1



- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner)

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.

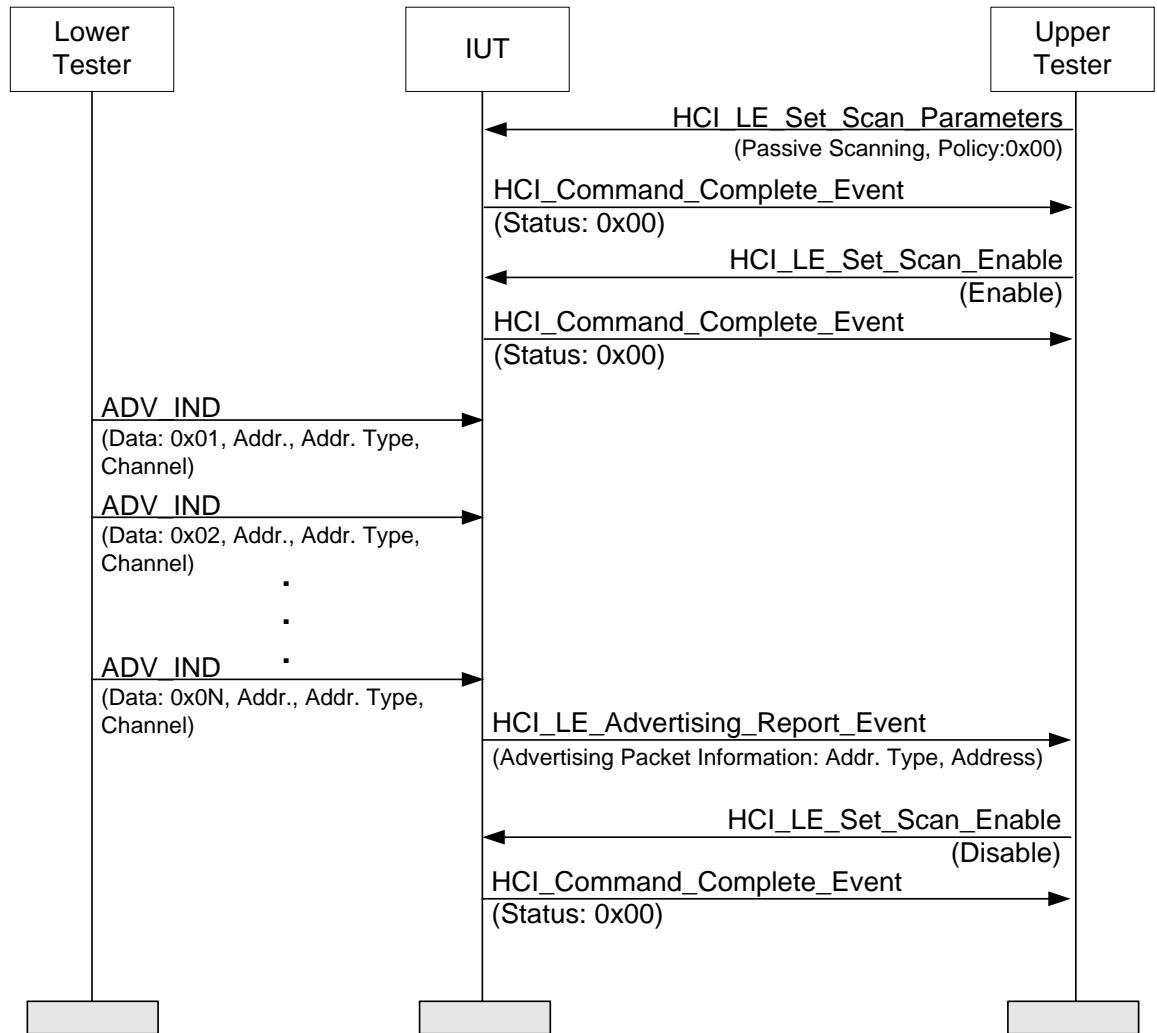


Figure 4.81: LL/DDI/SCN/BV-10-C [Passive Scanning: Undirected events]

- Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
- Upper Tester enables passive scanning in the IUT.
- Lower Tester sends ADV_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.



4. Upper Tester expects an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in Step 3 and data included in one of the advertising packets.
 5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 6. Configure Lower Tester as advertiser using the second supported advertising channel and an address with an identical company identifier but a different company assigned identifier than the IUT.
 7. Repeat steps 2-5.
 8. Configure Lower Tester as advertiser using the third supported advertising channel using an address with a different company identifier and using the same company assigned identifier as the IUT and a random address type.
 9. Repeat steps 2-5.
- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel separately, with the IUT recognizing the advertising packet type.

4.2.3.8 LL/DDI/SCN/BV-11-C [Passive Scanning: Directed Events]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using directed advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Device Address Set (supported type of address, any address) AND Passive Scanning (selected scan interval, selected scan window) AND Specific White Listed (policy for scanner)

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.



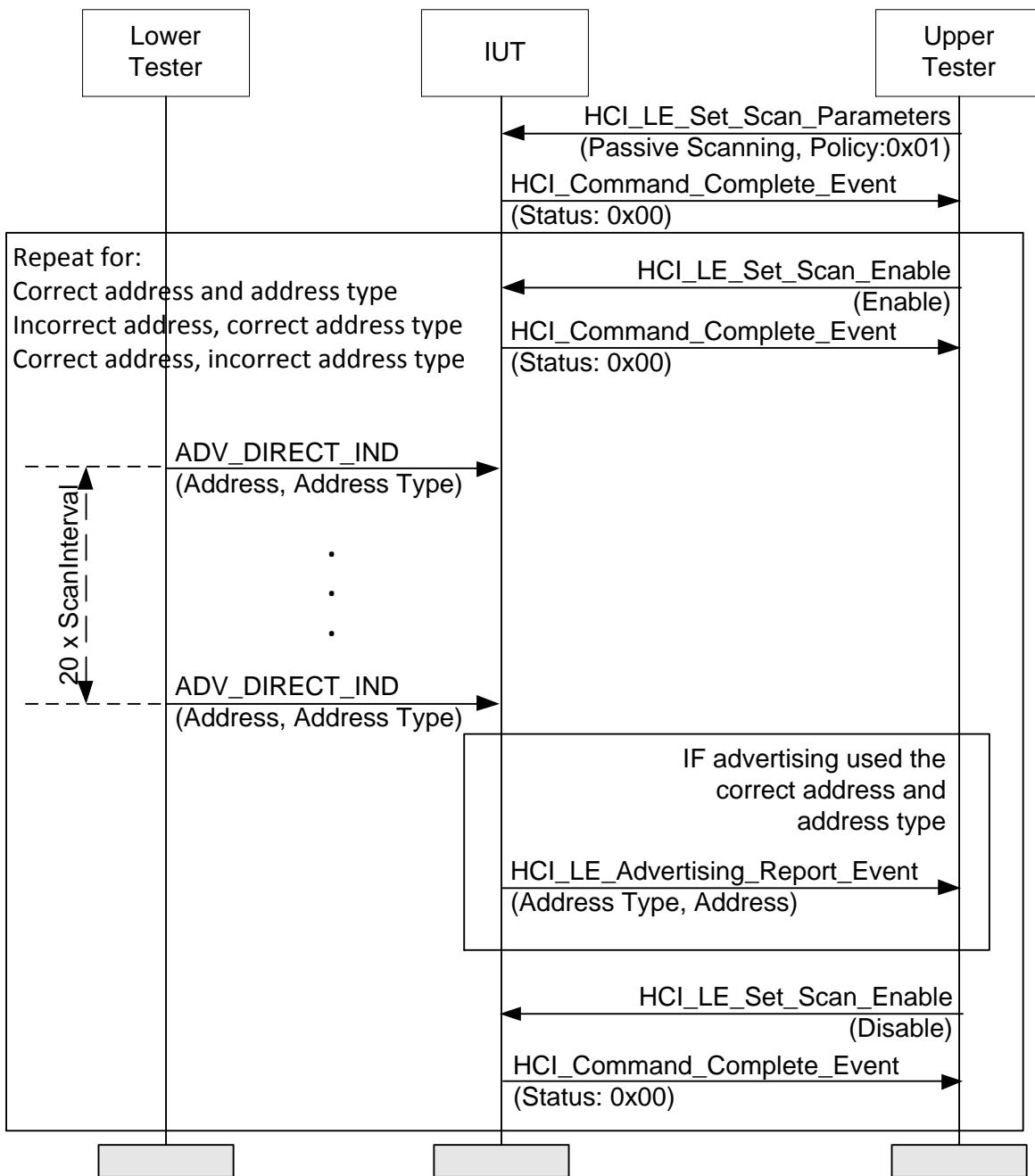


Figure 4.82: LL/DDI/SCN/BV-11-C [Passive Scanning: Directed events]

- Configure Lower Tester as advertiser using a first supported advertising channel, directed advertising with the TargetA set to the IUT's address, and the AdvA set to a common public address in the IUT's White List with a value that could be a valid random address (most significant bit is set to 0 or following bit is set to 1).
- Upper Tester enables passive scanning in the IUT.
- Lower Tester sends ADV_DIRECT_IND packets, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
- Upper Tester expects an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in Step 3.

5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 6. Configure Lower Tester as advertiser using the second supported advertising channel, directed advertising with the TargetA set to the IUT's address, and the AdvA set to a public address not in the IUT's White List.
 7. Repeat steps 2-5, except that in step 4, the Upper Tester expects not to receive an HCI_LE_Advertising_Report event.
 8. Configure Lower Tester as advertiser using the third supported advertising channel, directed advertising with the TargetA set to the IUT's address, and the AdvA set to a random address with the same 48-bit value as used in step 1.
 9. Repeat steps 2-5, except that in step 4, the Upper Tester expects not to receive an HCI_LE_Advertising_Report event.
- Expected Outcome

Pass Verdict

The test procedure completes using a specific advertising channel with the timing combination.

The IUT reports advertising for the device address matching the white listed address both for the address and type with an HCI event,

The IUT does not report the device addresses mismatching the white listed address with an HCI event.

The IUT does not report advertising for the device address matching the white listed address for the address but mismatching the type with an HCI event,

4.2.3.9 LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable Events]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets correctly.

The Lower Tester advertises using discoverable undirected advertising events on one channel at a time and expects the IUT to report the advertising by the Lower Tester. The advertising packets' payload is a sequence numbering of the packets.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Device Address Set (supported type of address, any address) AND Passive Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner)

- Test Procedure

Execute the test procedure using the minimum advertising interval, the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.



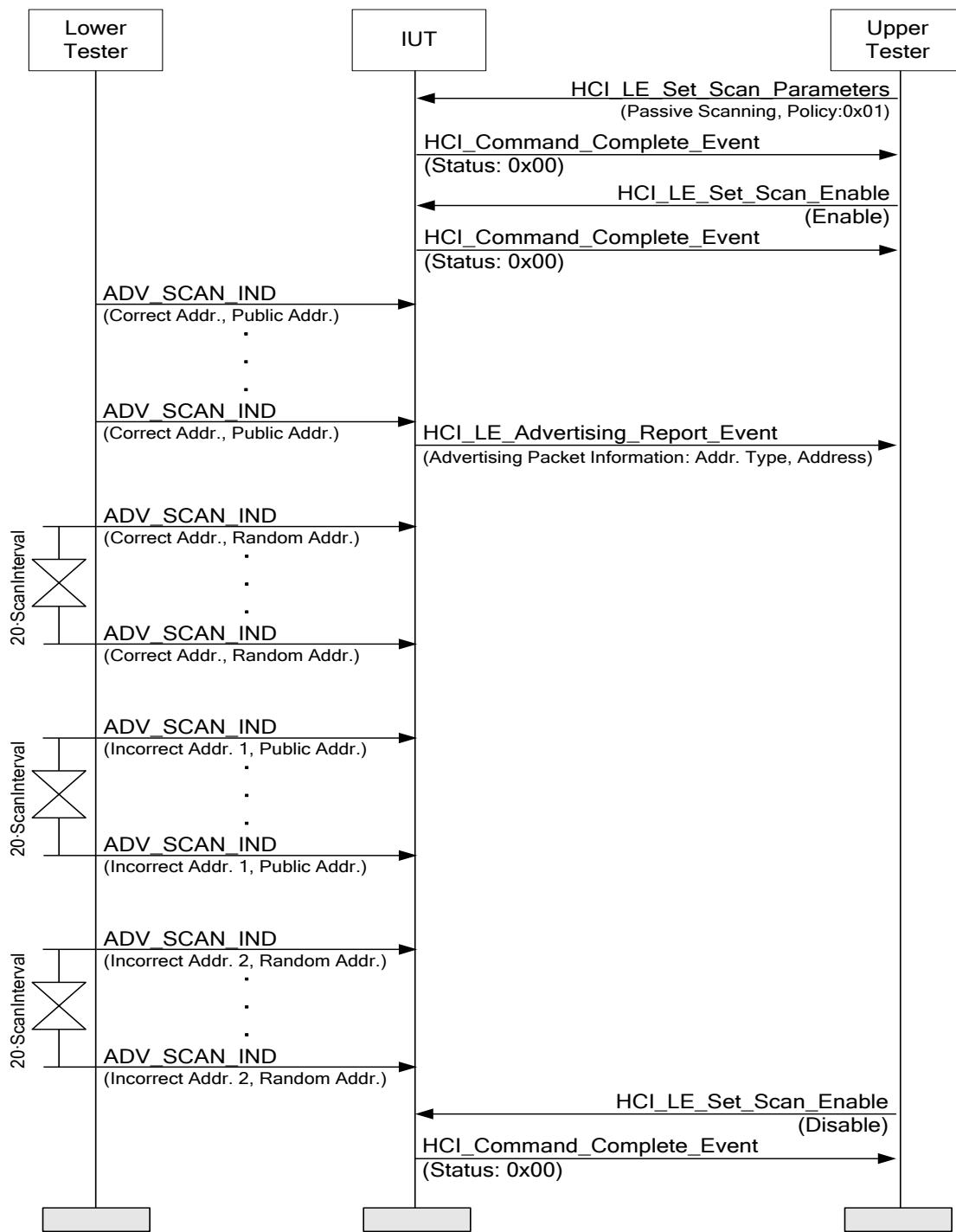


Figure 4.83: LL/DDI/SCN/BV-12-C [Passive Scanning: Discoverable events]

1. Configure Lower Tester as advertiser using a first supported advertising channel and a common public address.
2. Upper Tester enables passive scanning in the IUT.
3. Lower Tester sends ADV_SCAN_IND packets with the event count as data, each advertising event on the selected advertising channel only using the selected advertising interval until 20 scan intervals have passed.
4. Upper Tester expects an HCI_LE_Advertising_Report event from the IUT with advertising event type matching the type sent in Step 3 and data included in one of the advertising packets.

5. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 6. Configure Lower Tester as advertiser using the second supported advertising channel and an address with an identical company identifier but a different company assigned identifier than the IUT.
 7. Repeat steps 2-5.
 8. Configure Lower Tester as advertiser using the third supported advertising channel using an address with a different company identifier and using the same company assigned identifier as the IUT and a random address type.
 9. Repeat steps 2-5.
- Expected Outcome

Pass Verdict

The test procedure completes using a specific advertising channel with the timing combination.

The IUT reports the device address matching the white listed both for the address and type with an HCI event,

The IUT does not report the device address matching the white listed for the address and but mismatching the type with an HCI event,

The IUT does not report the device addresses mismatching the white listed address with an HCI event.

4.2.3.10 LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]

- Test Purpose

Verify the IUT when doing passive scanning and using the Resolving List reports advertising from the Lower Tester. The Lower Tester is doing non-connectable advertising and uses a resolvable private address for the AdvA field, i.e. the Lower Tester has distributed its own IRK.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertising packets (0x00)).

The IUT is using a resolvable private address (0x02 or 0x03).



- Test Procedure

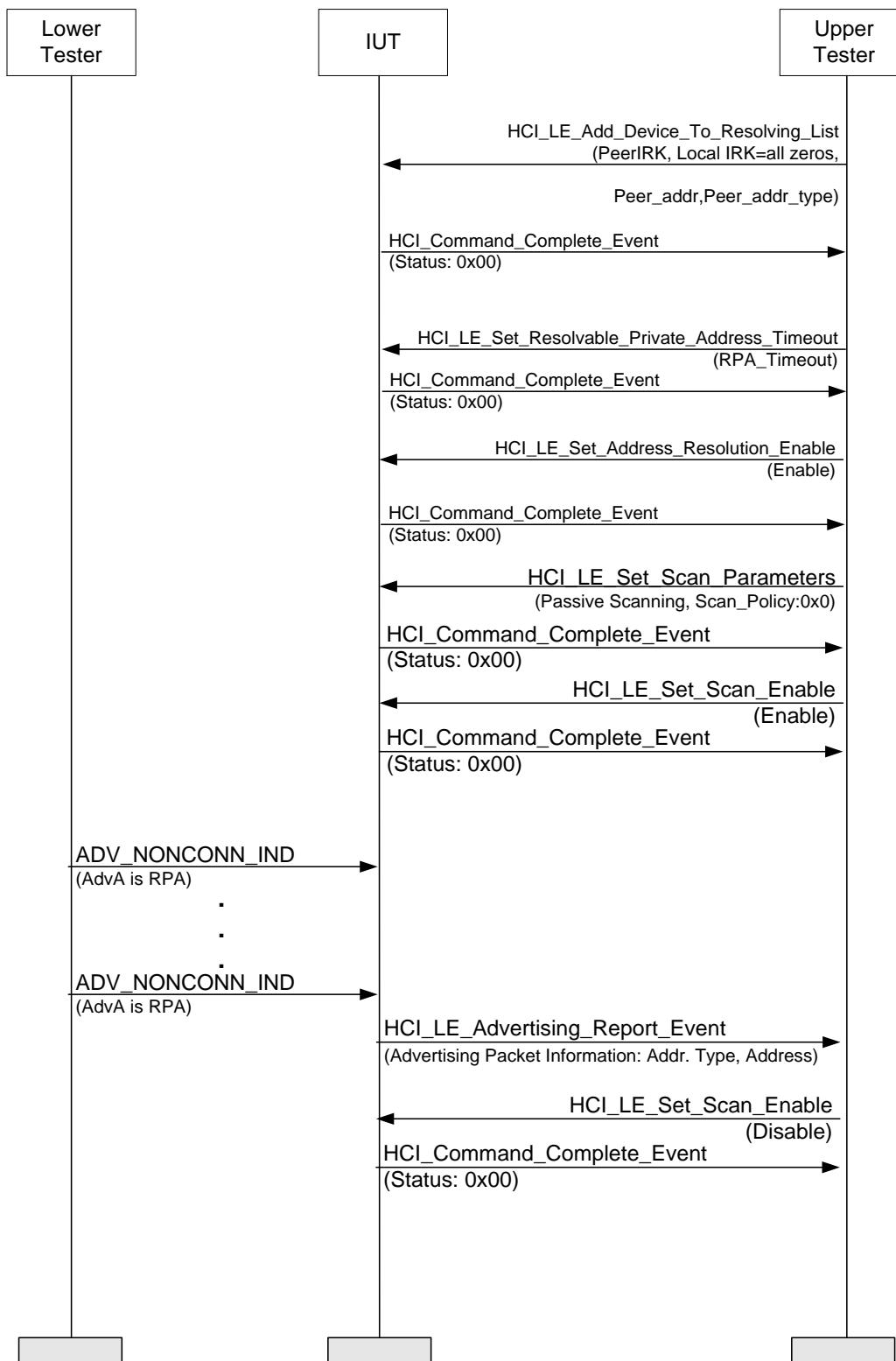


Figure 4.84: LL/DDI/SCN/BV-13-C [Network Privacy – Passive Scanning, Peer IRK]

1. The Upper Tester populates the IUT resolving list with the peer IRK and identity address.
2. The Upper Tester enables passive scanning in the IUT.



3. Configure the Lower Tester to start advertising. The Lower Tester uses a resolvable private address in the AdvA field.
 4. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
 5. The Upper Tester expects at least one HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester. The address in the report is resolved by the IUT using the distributed IRK.
 6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The IUT receives and reports advertising with a resolvable private address in the AdvA field from the Lower Tester.

4.2.3.11 LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning: Directed Events to an address different from the scanner's address]

- Test Purpose

Verify that a scanner IUT detects and reports directed advertising packets correctly, when the Upper Tester has set the scan filter policy to 0x02. The Upper Tester sets the own address to a resolvable private address, the Lower Tester sends connectable directed advertising packets to the IUT with InitA set to a resolvable private address that is different from the address that the Upper Tester has provided to the IUT. The IUT reports the directed advertising packets to the Upper Tester.

- Reference

[3] 4.4.3.1, 4.3.3

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND extended filtering policy to process all advertising packets (0x02)

The IUT is using a resolvable private address (0x02 or 0x03).

The Lower Tester is also using a resolvable private address.



- Test Procedure

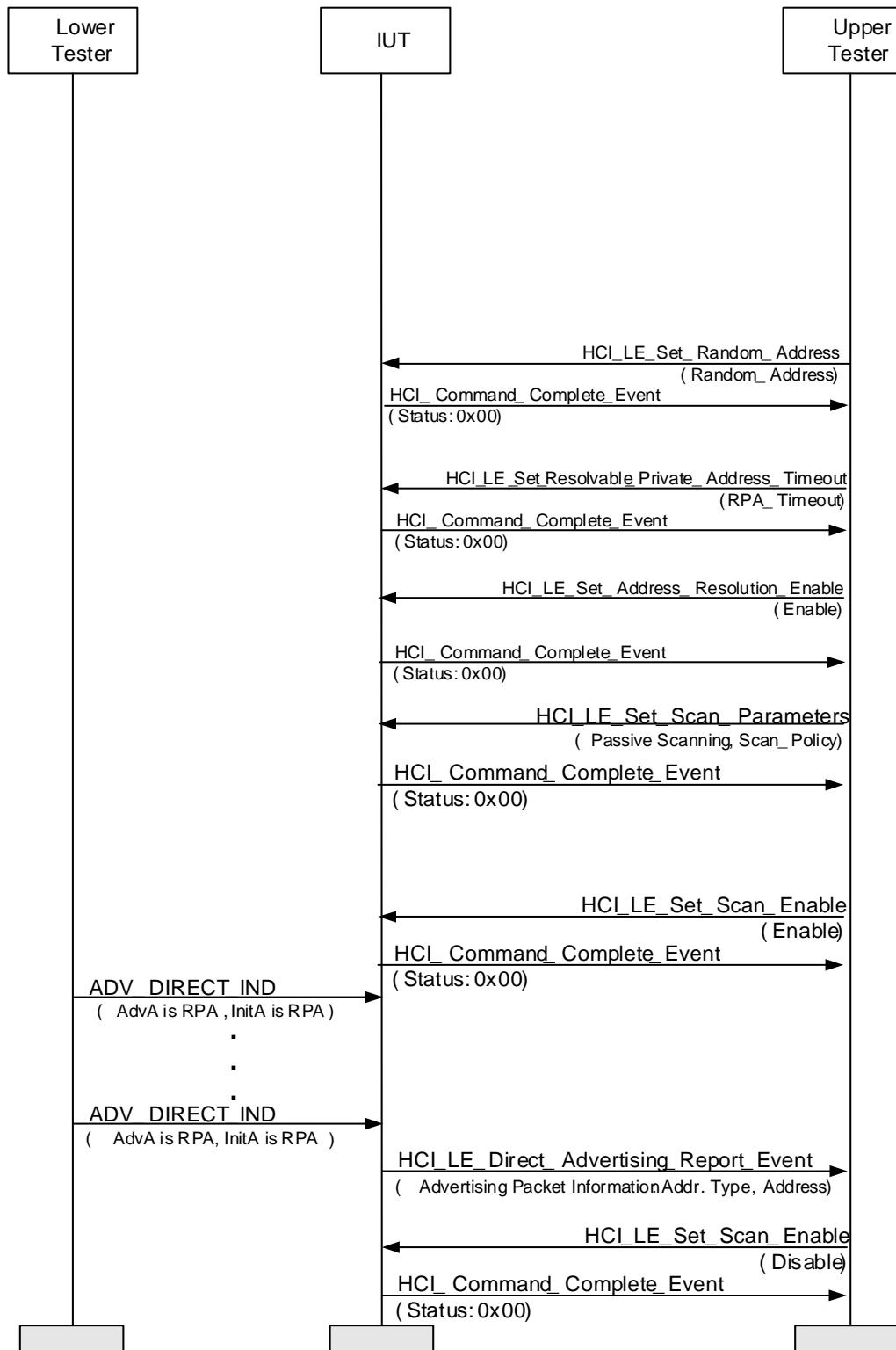


Figure 4.85: LL/DDI/SCN/BV-14-C [Network Privacy - Passive Scanning: Directed Events to an address different from the scanner's address]

1. The Upper Tester sets a resolvable private address for the IUT to use.
2. The Upper Tester enables passive scanning using filter policy 0x02 in the IUT.

- 3. Configure the Lower Tester to start advertising. The Lower Tester uses a resolvable private address type in the AdvA field. The InitA field also contains a resolvable private address, which does not match the address set by the Upper Tester in the IUT.
- 4. The Lower Tester sends an ADV_DIRECT_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
- 5. The Upper Tester expects at least one HCI_LE_Direct_Advertising_Report reporting the advertising packets sent by the Lower Tester.
- 6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The IUT receives directed advertising with a resolvable private address different from the IUT address in the InitA field from the Lower Tester. The advertising packets are reported to the Upper Tester.

4.2.3.12 LL/DDI/SCN/BV-15-C [Network Privacy – Active Scanning, no Local IRK, no Peer IRK]

- Test Purpose

Verify that the IUT when doing active scanning reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a non-resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising. The Lower Tester has not distributed its own IRK.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
 LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
 LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window)
 AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))



- Test Procedure

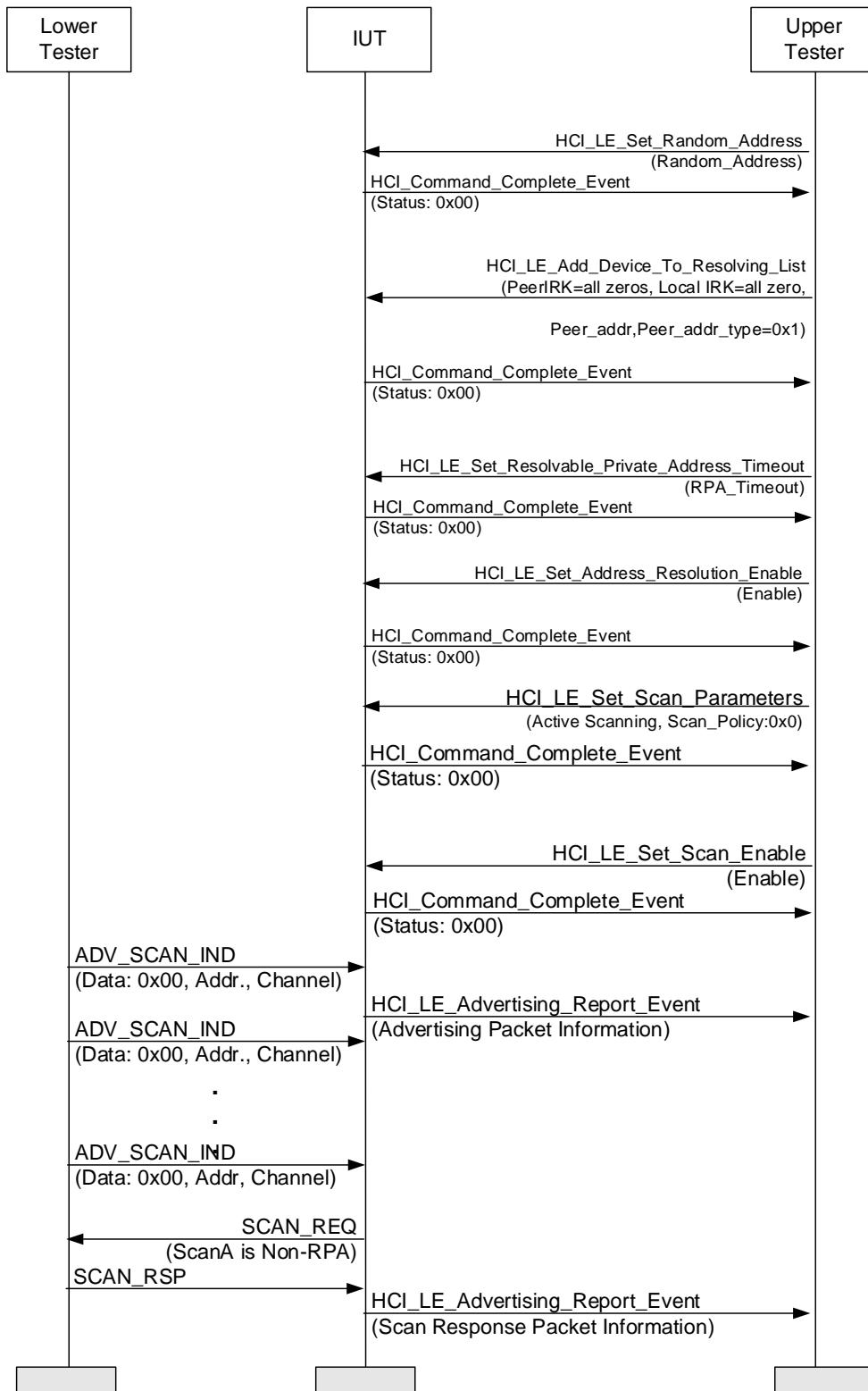


Figure 4.86: LL/DDI/SCN/BV-15-C [Network Privacy – Active Scanning, no Local IRK, no Peer IRK]

1. Configure the Lower Tester to start advertising.
 2. Configure the Lower Tester as an advertiser using a private address (either resolvable or non-resolvable) in the AdvA field.

3. Upper Tester enables address resolution and active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.
 4. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 5 occurs.
 5. Lower Tester expects a SCAN_REQ packet T_IFS after any of the ADV_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address T_IFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a non-resolvable private address.
 6. Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
 7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The IUT receives undirected advertising using a private address in the AdvA field from the Lower Tester.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a non-resolvable private address in the ScanA field.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.13 LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK]

- Test Purpose

Verify that the IUT, when doing active scanning and using the Resolving List, reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a public or static random address for the AdvA field. The Lower Tester has not distributed its own IRK.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))



- Test Procedure

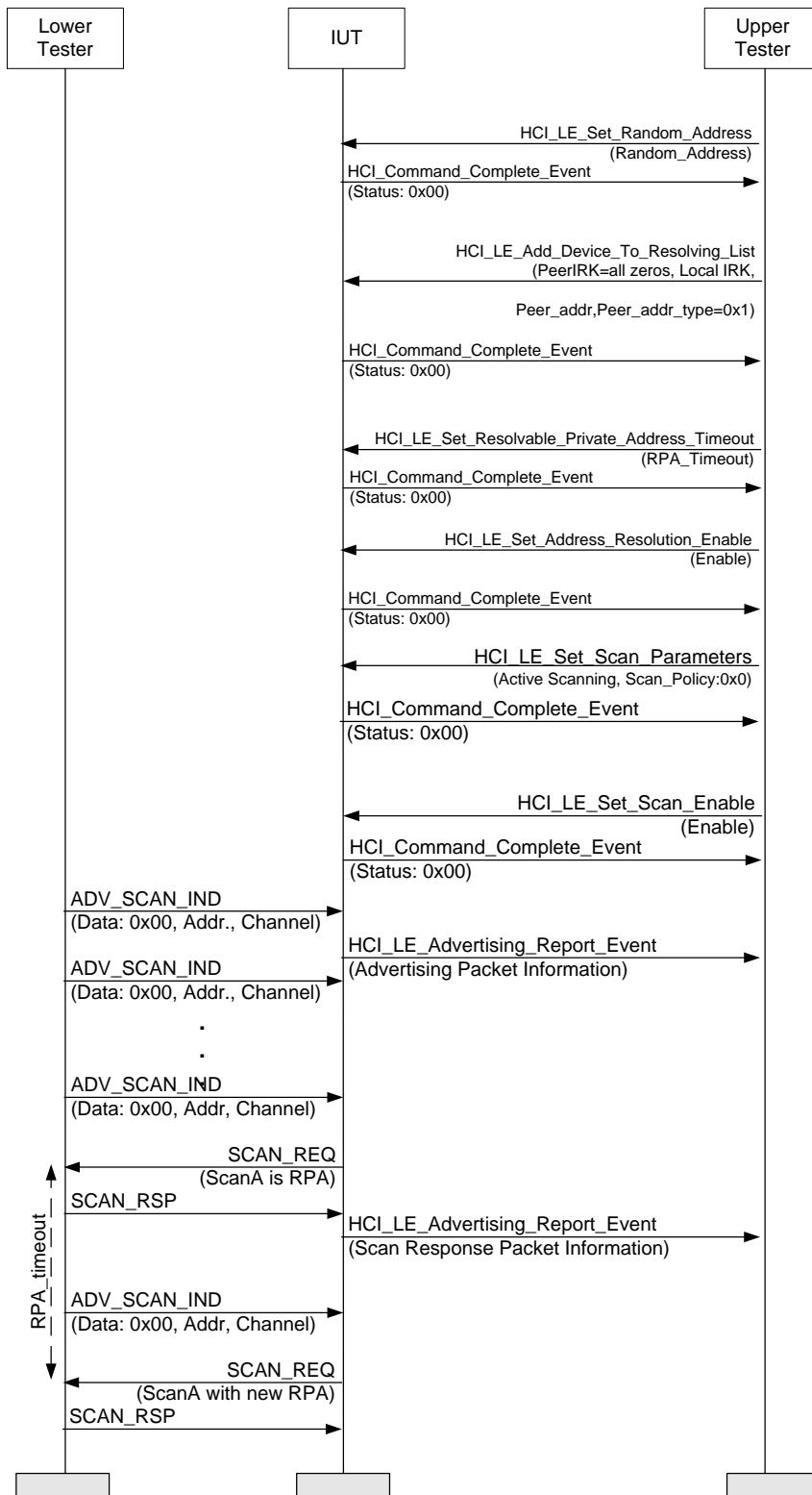


Figure 4.87: LL/DDI/SCN/BV-16-C [Network Privacy – Active Scanning, Local IRK, no Peer IRK]

- Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a random static address.
- Configure the Lower Tester as an advertiser using its device identity address in the AdvA field.
- Upper Tester adds the peer device to the resolving list with a valid local IRK and no peer IRK.



4. Upper Tester enables active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.
5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 6 completes.
6. Lower Tester expects a SCAN_REQ packet TIFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address TIFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a resolvable private address.
7. Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
8. Wait for address refresh timeout starting from the first SCAN_REQ received and repeat steps 5 and 6 to verify that the ScanA field in the SCAN_REQ packet has now changed to a new resolvable private address.
9. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.

- Expected Outcome

Pass Verdict

The IUT receives undirected advertising using the Lower Tester's Identity Address in the AdvA field from the Lower Tester.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The ScanA field should contain a new resolvable private address after the address refresh timeout.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.14 LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK]

- Test Purpose

Verify that the IUT, when doing active scanning and using the Resolving List reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a non-resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a resolvable private address for the AdvA field. The Lower Tester has distributed its own IRK.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))



- Test Procedure

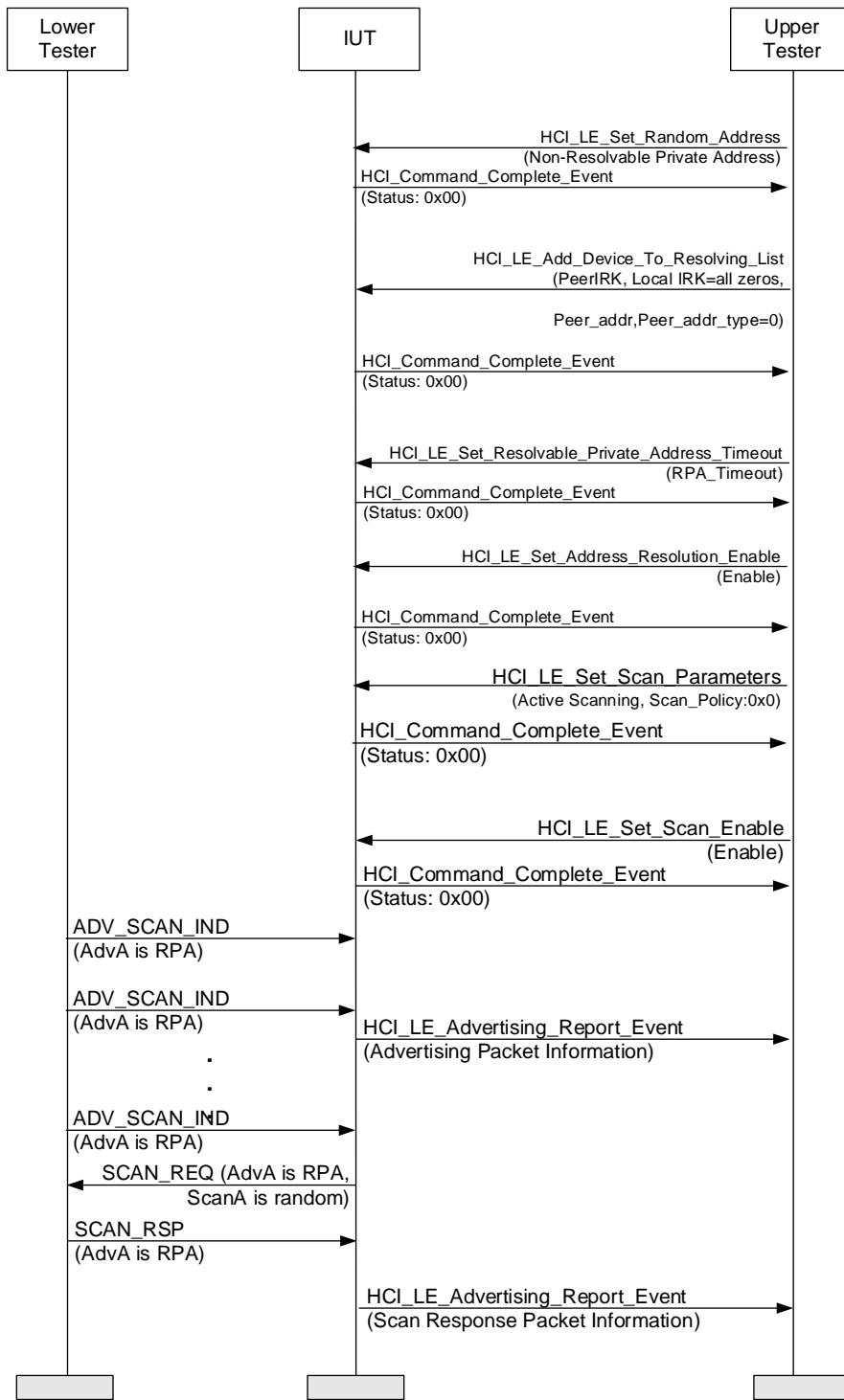


Figure 4.88: LL/DDI/SCN/BV-17-C [Network Privacy – Active Scanning, no Local IRK, Peer IRK]

1. Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a non-resolvable private address.
 2. Configure the Lower Tester advertising channel map using the first supported advertising channel.
 3. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.

4. Upper Tester adds peer identity and IRK information to resolving list.
 5. Upper Tester enables active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.
 6. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
 7. Lower Tester expects a SCAN_REQ packet TIFS after any of the ADV_SCAN_IND packets. Lower Tester sends a SCAN_RSP packet containing the selected address TIFS after the SCAN_REQ packet. The ScanA field in the SCAN_REQ packet should use a non-resolvable private address.
 8. Upper Tester expects an HCI_LE_Advertising_Report event containing the ADV_SCAN_IND response information sent in step 7.
- Expected Outcome

Pass Verdict

The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a non-resolvable private address in the ScanA field.

4.2.3.15 LL/DDI/SCN/BV-18-C [Network Privacy – Active Scanning, Local IRK, Peer IRK]

- Test Purpose

Verify that the IUT when doing active scanning and using the Resolving List reports the Lower Tester and sends SCAN_REQs to the Lower Tester with a resolvable private address for the ScanA field. The Lower Tester is doing scannable undirected advertising and uses a resolvable private address for the AdvA field i.e. the IUT and Lower Tester has distributed its Device Identities.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))



- Test Procedure

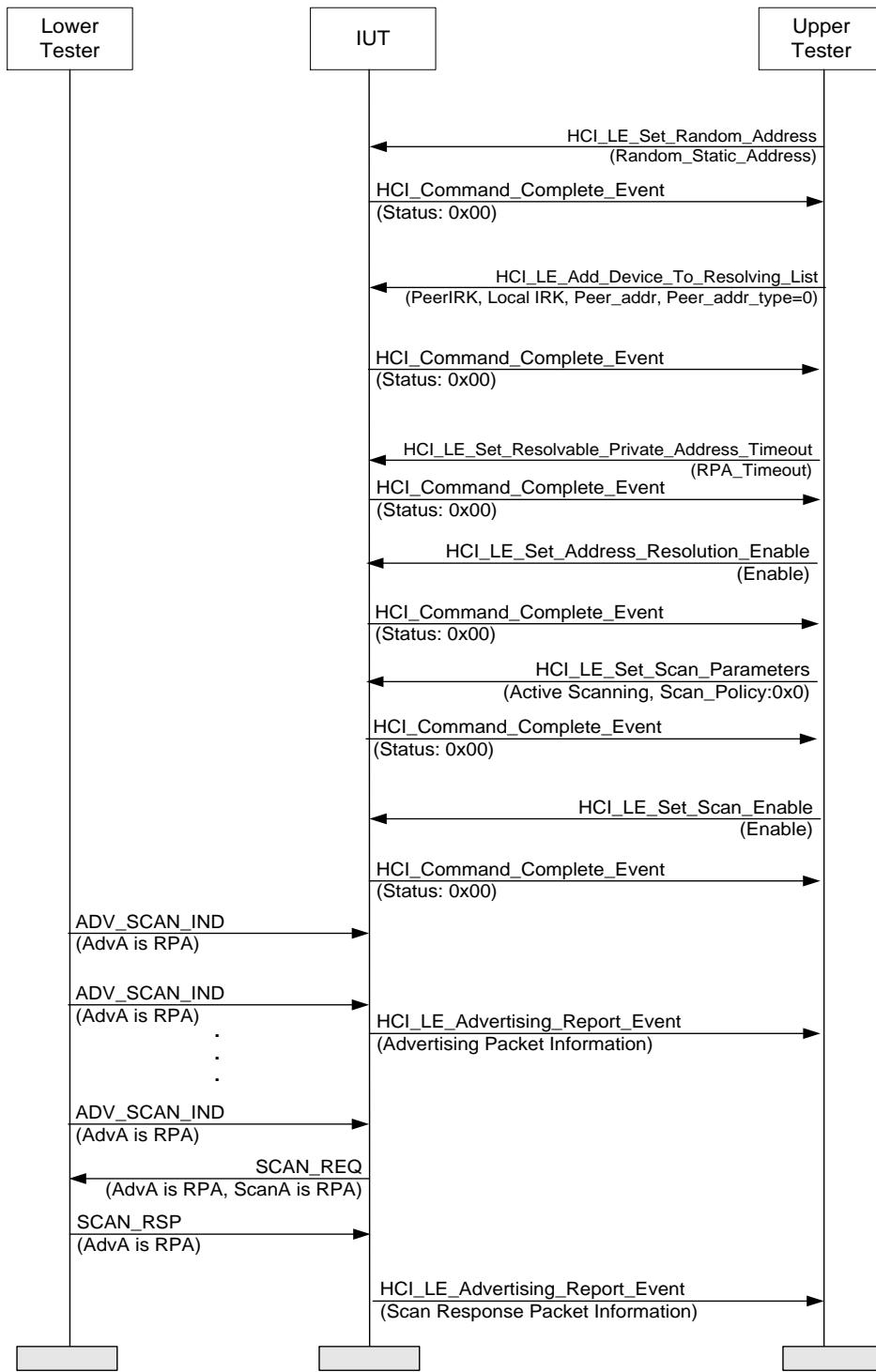


Figure 4.89: LL/DDI/SCN/BV-18-C [Network Privacy – Active Scanning, Local IRK, Peer IRK]

1. Upper Tester sends an HCI_LE_Set_Random_Address to the IUT with a random static address.
 2. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.
 3. Upper Tester adds peer device identity and local IRK information to resolving list.
 4. Upper Tester enables active scanning with filtering policy set to 'Accept all advertising packets (0x00)' in the IUT.

5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
 6. Lower Tester expects a SCAN_REQ packet T_IFS after any of the ADV_SCAN_IND packets. The ScanA field in the SCAN_REQ packet shall use the same resolvable private address.
 7. Lower Tester sends a SCAN_RSP packet T_IFS after the SCAN_REQ packet. The AdvA field in the SCAN_RSP packet should use the resolvable private address that was used in the SCAN_REQ packet.
 8. Interleave with step 6: Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
 9. Interleave with step 7: Upper Tester expects an HCI_LE_Advertising_Report event containing the scan response information.
- Expected Outcome

Pass Verdict

The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester. The IUT successfully resolves the address.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The Lower Tester successfully resolves the address.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.2.3.16 LL/DDI/SCN/BV-19-C [Extended Scanning, Passive]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets received correctly, including legacy and extended PDUs. The Lower Tester advertises using non-connectable and non-scannable advertising events on the primary and secondary advertising channels and expects the IUT to report the advertising to the Upper Tester. Both directed and undirected advertising events are tested, with and without data.

- Reference

[10] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- Test Procedure

Execute the test procedure using the minimum advertising interval and continuous passive scanning.



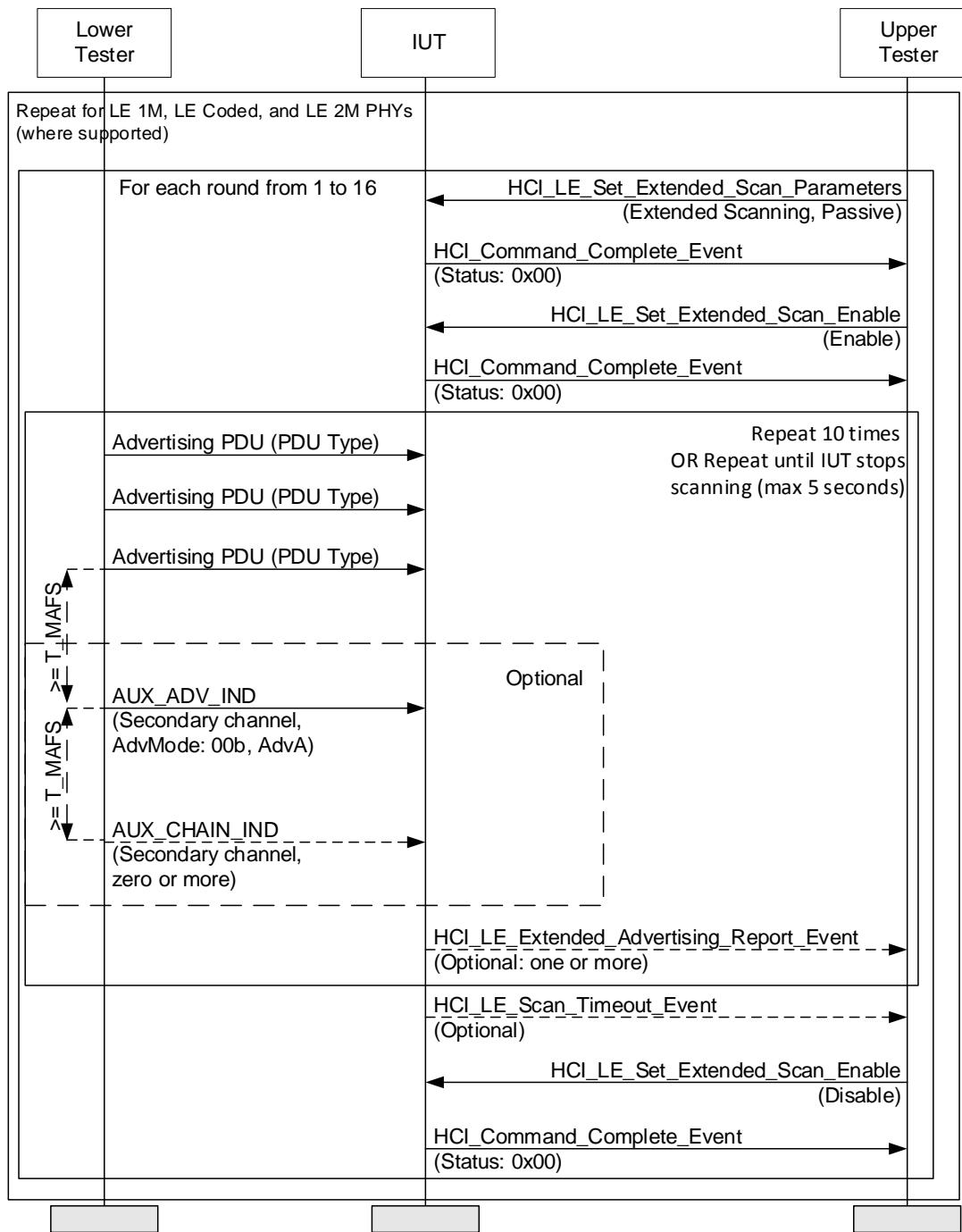


Figure 4.90: LL/DDI/SCN/BV-19-C [Extended Scanning, Passive]

For each round from 1 to 16 based on [Table 4.19](#):

1. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Parameters_Command` to the IUT. The `Scanning_PHYs` parameter shall be set to 0x01 (LE 1M), `Scan_Type[0]` set to 0x00 (Passive Scanning), `Scan_Interval[0]` set to 0x0010, and `Scan_Window[0]` set to 0x0010. `Own_Address_Type` shall be set to 0x00 (Public Device Address), and `Scanning_Filter_Policy` shall be set to 0x00 (Accept All).
2. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Enable` command to the IUT to enable scanning. `Filter_Duplicates` and `Period` shall be set to zero. The `Duration` parameter shall be set to the value specified in [Table 4.19](#) for this round.

3. The Lower Tester begins advertising on the LE 1M PHY using the PDU Type specified in [Table 4.19](#) for this round. If AUX_ADV_IND is included in the round, the ADV_EXT_IND shall include an AuxPtr that refers to the AUX_ADV_IND on the LE 1M PHY, and all fields specified should be included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where "LT" equals the Lower Tester address. If InitA/TargetA is specified the appropriate PDU shall include the field, where "IUT" equals the IUT address and "Not IUT" equals a random address other than the IUT address. If AdvData is specified the PDU shall include the field populated with random octets of the specified count. If the AdvData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data. If Duration is set to 0x0000, repeat for at least 20 advertising intervals, otherwise repeat until the end of the round.
4. For undirected advertisements or advertisements directed at the IUT, the Upper Tester expects one or more HCI_LE_Extended_Advertising_Report events from the IUT with an advertising event type matching the type sent in Step 3 and the Primary_PHY set to 0x01 (LE 1M), and if the advertisements used extended PDUs, the Secondary_PHY shall be set to 0x01 (LE 1M). If AdvData was included in the advertisement, the Upper Tester expects to receive the data included in one or more of the advertising packets. If the advertisement was directed at the IUT, the Upper Tester expects to receive the Direct Address Type and Direct Address used to direct the advertisement at the IUT.
5. If the Duration was set to 0x0000 (No Scanning Duration), repeat step 4 until a number of advertising reports (10) have been generated. Each time the Upper Tester receives a report, the Lower Tester shall change the AdvData, if any. If the round uses extended advertising PDUs, it shall also change the DID sub-field of the ADI field to a new value. Otherwise repeat step 4 until the amount of time specified for Duration has elapsed. Afterwards, the Upper Tester expects an HCI_Scan_Timeout event from the IUT. Skip step 6.
6. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
7. Repeat steps 1-5 for each Round shown in [Table 4.19](#).
8. If the IUT supports the LE Coded PHY, repeat steps 1-7, except that in Step 1 the Scanning_PHYs shall be set to 0x04 (LE Coded), in Step 3 the Lower Tester advertises using the LE Coded PHY for both primary and secondary advertising channels, in Step 4 the Primary_PHY and Secondary_PHY shall both be set to 0x03 (LE Coded), and in Step 7 only rounds 10-16 in [Table 4.19](#) are executed.
9. If the IUT supports the LE 2M PHY, repeat steps 1-7, except that in Step 1 the Scanning_PHYs shall be set to 0x01 (LE 1M), in Step 3 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel, in Step 4 Primary_PHY shall be set to 0x01 (LE 1M) and Secondary_PHY shall be set to 0x02 (LE 2M) , and in Step 7 only rounds 10-16 in [Table 4.19](#) are executed.

Note: For rounds 13 and 14 from [Table 4.19](#), if the IXIT value for "Scan Max Data" is less than or equal to the AdvData length shown in the table, then omit this round.

Round	LE Set Extended Scan Enable (Step 2)	Lower Tester Advertising PDUs			
	Duration	PDU Type (AdvMode)	AdvA	InitA or TargetA	AdvData
1	0x0000	ADV_IND	LT	-	None
2	0x0000	ADV_IND	LT	-	31 octets



Round	LE Set Extended Scan Enable (Step 2)	Lower Tester Advertising PDUs			
	Duration	PDU Type (AdvMode)	AdvA	InitA or TargetA	AdvData
3	0x0000	ADV_DIRECT_IND	LT	IUT	-
4	0x0000	ADV_DIRECT_IND	LT	Not IUT	-
5	0x0000	ADV_NONCONN_IND	LT	-	None
6	0x0000	ADV_NONCONN_IND	LT	-	31 octets
7	0x0000	ADV_EXT_IND (00b)	LT	-	-
8	0x0000	ADV_EXT_IND (00b)	LT	IUT	-
9	0x0000	ADV_EXT_IND (00b)	LT	Not IUT	-
10	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	-	- None
11	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	- IUT	- None
12	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	- Not IUT	- None
13	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	-	- 191
14	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b) AUX_CHAIN_IND (00b)	- LT	-	- 382
15	0x0000	ADV_EXT_IND (00b) AUX_ADV_IND (00b) AUX_CHAIN_IND (00b)	- LT	-	- Scan_Max_Data
16	0x01F4 (5 s)	ADV_EXT_IND (00b)	LT	IUT	-

Table 4.19: PDU payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

- For undirected advertisements or advertisements directed at the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events with an advertising event type matching the type sent in Step 3 and the Primary_PHY set correctly, and if the advertisements used extended PDUs, the Secondary_PHY shall be set correctly.
- If AdvData was included in the advertisement, the Upper Tester expects to receive the correct data in the Data field, with Data_Length set correctly, in one or more HCI_LE_Extended_Advertising_Report events. The Upper Tester expects to receive at least one new event each time the AdvData or DID changes.
- If the advertisement was directed at the IUT, the Upper Tester expects to receive the Direct Address Type and Direct Address used to direct the advertisement at the IUT.



- For directed advertisements not directed to the IUT, the IUT does NOT generate an HCI_LE_Extended_Advertising_Report event or receive the data.
- If a nonzero Duration was specified, the IUT generates an LE Scan Timeout event after the duration has elapsed and stops scanning.

4.2.3.17 LL/DDI/SCN/BV-20-C [Extended Scanning, Active]

- Test Purpose

Tests that a scanner IUT detects and requests additional information from advertisements received and reports the results from the Controller. The Lower Tester advertises using scannable extended advertising events on one channel at a time and expects the IUT to report the advertising to the Upper Tester. Both directed and undirected advertising events are tested.

- Reference

[10] 4.4.3.2

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (Active, selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- Test Procedure

Execute the test procedure using the minimum advertising interval and continuous active scanning



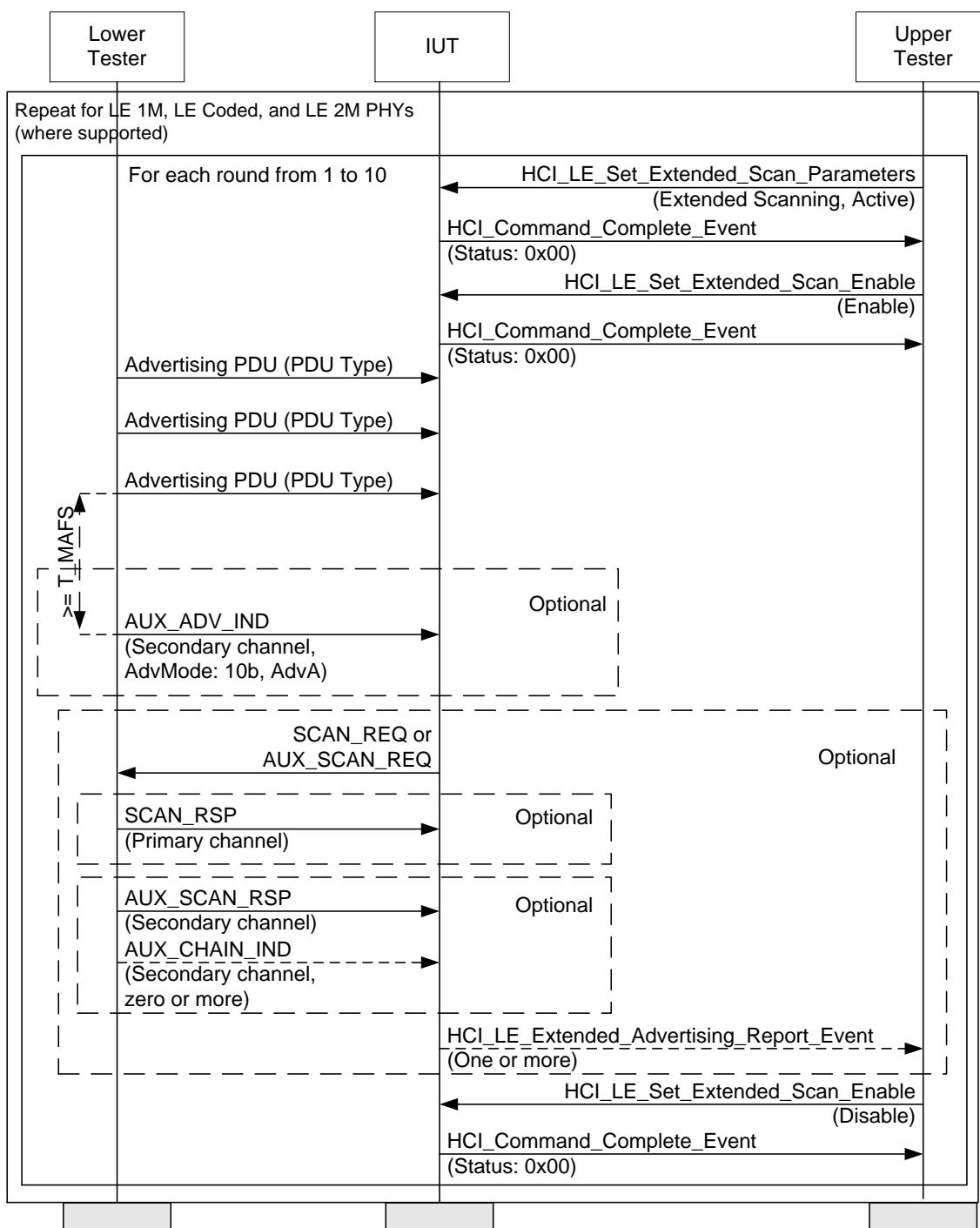


Figure 4.91: LL/DDI/SCN/BV-20-C [Extended Scanning, Active]

- For each round from 1 to 10 based on [Table 4.20](#), if ScanData Length is less than or equal to the "Scan Max Data" then perform Steps 2 to 8 and otherwise omit this round.
- The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Parameters** command to the IUT. The **Scanning_PHYs** parameter shall be set to 0x01 (LE 1M), **Scan_Type[0]** set to 0x01 (Active Scanning), **Scan_Interval[0]** set to 0x0010, and **Scan_Window[0]** set to 0x0010. **Own_Address_Type** shall be set to 0x00 (Public Device Address), and **Scanning_Filter_Policy** shall be set to 0x00 (Accept All).
- The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Enable** command to the IUT to enable scanning. **Filter_Duplicates**, **Duration**, and **Period** shall all be set to zero.

4. The Lower Tester begins advertising on the LE 1M PHY using the PDU Type specified in [Table 4.20](#) for this round. If AUX_ADV_IND is included in the round, the ADV_EXT_IND shall include an AuxPtr that refers to the AUX_ADV_IND on the LE 1M PHY, and all fields specified should be included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where "LT" equals the Lower Tester address. If TargetA is specified the appropriate PDU shall include the field, where "IUT" equals the IUT address and "Not IUT" equals a random address other than the IUT address. Repeat for at least 20 advertising intervals or until Step 5 occurs.
5. For undirected advertisements or advertisements directed at the IUT, the Lower Tester expects to receive either a SCAN_REQ (if advertising with legacy PDUs) or an AUX_SCAN_REQ (if advertising with extended PDUs) on the appropriate advertising channel. The ScanA field shall be set to the IUT's address and the AdvA address set to the Lower Tester's address. If the advertisement was directed at the IUT, the Upper Tester expects to receive the Direct Address Type and Direct Address used to direct the advertisement at the IUT. If the advertisements were not directed at the IUT, skip to Step 8.
6. If the IUT sent a SCAN_REQ in Step 5, the Lower Tester responds with an SCAN_RSP packet to the IUT TIFS after the end of the SCAN_REQ PDU. If ScanData is specified, the SCAN_RSP PDU shall include the field populated with random octets from 1 to 255, of the specified count.
OR
If the IUT sent an AUX_SCAN_REQ in Step 5, the Lower Tester responds with an AUX_SCAN_RSP packet to the IUT TIFS after the end of the AUX_SCAN_REQ PDU with an AdvMode of 00b. If ScanData is specified, the AUX_SCAN_RSP PDU shall include the field populated with random octets from 1 to 255, of the specified count. If the ScanData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data.
7. If the Lower Tester sent a scan response in Step 6, the Upper Tester expects one or more HCI_LE_Extended_Advertising_Report events from the IUT with an Event_Type where bit 3 (Scan response) is set. If ScanData was included in the response, the Upper Tester expects to receive the data included in one of the advertising packets.
8. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
9. If the IUT supports the LE Coded PHY, repeat Steps 1-8, except that in Step 1 only rounds 5-8 in [Table 4.20](#) are executed, in Step 2 the Scanning_PHYs shall be set to 0x04 (LE Coded), and in Step 4 the Lower Tester advertises using the LE Coded PHY for both primary and secondary advertising channels.
10. If the IUT supports the LE 2M PHY, repeat Steps 1-8, except that in Step 1 only rounds 5-8 in [Table 4.20](#) are executed, in Step 2 the Scanning_PHYs shall be set to 0x01 (LE 1M), and in Step 4 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel.

Round	Lower Tester Advertising PDUs			
	PDU Type (AdvMode)	AdvA	TargetA	ScanData
1	ADV_IND	LT	-	None
2	ADV_IND	LT	-	31 octets
3	ADV_SCAN_IND	LT	-	None
4	ADV_SCAN_IND	LT	-	31 octets
5	ADV_EXT_IND (10b) AUX_ADV_IND (10b)	- LT	- -	- 1



Round	Lower Tester Advertising PDUs				
	PDU Type (AdvMode)	AdvA	TargetA	ScanData	
6	ADV_EXT_IND (10b)	-	-	-	
	AUX_ADV_IND (10b)	LT	IUT	1	
7	ADV_EXT_IND (10b)	-	-	-	
	AUX_ADV_IND (10b)	LT	Not IUT	1	
8	ADV_EXT_IND (10b)	-	-	-	
	AUX_ADV_IND (10b)	LT	-	191	
9	ADV_EXT_IND (10b)	-	-	-	
	AUX_ADV_IND (10b)	LT	-	382	
	AUX_CHAIN_IND (00b)	-	-		
10	ADV_EXT_IND (10b)	-	-	-	
	AUX_ADV_IND (10b)	LT	-		Scan_Max_Data
	AUX_CHAIN_IND (00b)	-	-		

Table 4.20: PDU payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

- For undirected advertisements or advertisements directed at the IUT, the IUT sends a scan request PDU to the LT.
- For scan responses received by the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events with an advertising event type matching the type sent in Step 4. If ScanData was included in the advertisement, the Upper Tester expects to receive the correct data in the Data field, with Data_Length set correctly.
- If the advertisement was directed at the IUT, the Upper Tester expects to receive the Direct Address Type and Direct Address used to direct the advertisement at the IUT.
- For directed advertisements not directed to the IUT, the IUT does NOT send a scan request or generate an HCI_LE_Extended_Advertising_Report event.

4.2.3.18 LL/DDI/SCN/BV-21-C [Extended Scanning, Periodic Advertising Reception]

- Test Purpose

Tests that a scanner IUT can locate and receive periodic advertising events and reports the results from the Controller. The Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising and expects the IUT to report the periodic advertising data by the Upper Tester. Confirms the IUT receives periodic advertisement PDUs sent using the Channel Selection Algorithm #2.

- Reference

[10] 4.4.2.13.1, 4.4.3.4



- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.



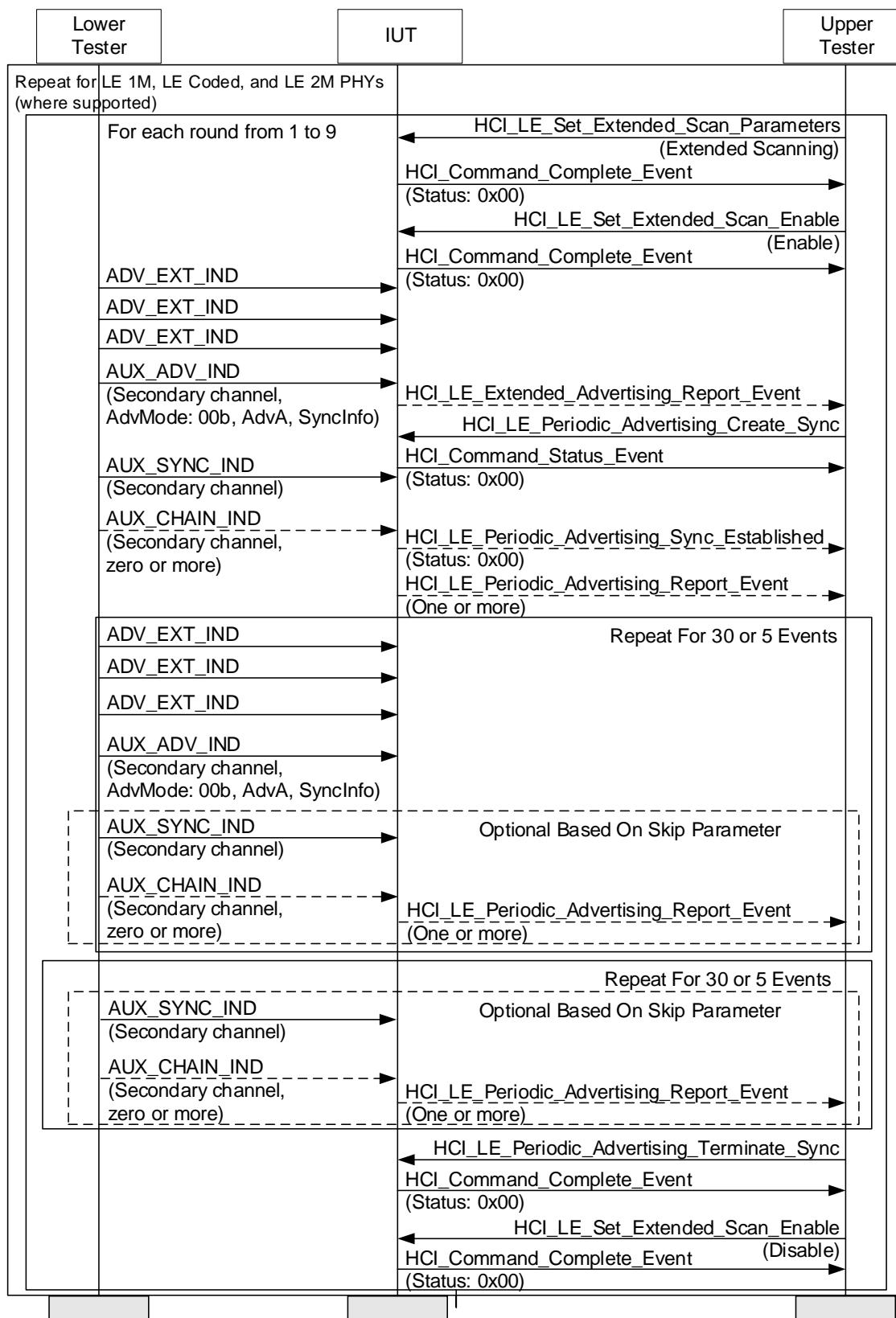


Figure 4.92: LL/DDI/SCN/BV-21-C [Extended Scanning, Periodic Advertising Reception]



For each round from 1 to 8 based on [Table 4.21](#):

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising using ADV_EXT_IND and AUX_ADV_IND PDUs using the LE 1M PHY. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
4. The Lower Tester generates ADV_SYNC_IND PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in Step 3. If the Data Length column in [Table 4.21](#) for this Round is non-zero, the AUX_SYNC_IND PDUs shall contain an AdvData field where the first 2 octets contain a 16-bit counter which is initially 0 and is incremented at each periodic event, and the remainder contain random octets, up to Data Length in [Table 4.21](#) or the maximum that can fit in the PDU. If all the data cannot fit in a single AUX_SYNC_IND PDU, the Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs containing the remaining data. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
5. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval.
6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Filter_Policy shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Skip shall be set to the Skip value in [Table 4.21](#). Sync_Timeout shall be set to (Skip + 3) x Periodic_Advertising_Interval from Step 5. Unused shall be set to 0x00. The Upper Tester expects an HCI_Command_Status event in response.
7. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
8. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. If AdvData is being advertised, the Upper Tester expects to receive the data included in the advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to 0xFF.
9. Repeat Step 8 until at least 30 HCI_LE_Periodic_Advertising_Report events if current round is round 7, and 5 events for other rounds have been received.
10. The Lower Tester ceases extended advertising but continues periodic advertising.
11. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report events from the IUT as in Step 8.
12. Repeat Step 11 until at least 30 events have been received if current round is round 7, and until at least 5 events have been received for other rounds.
13. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
14. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.



15. The Lower Tester ceases periodic advertising.
16. Repeat steps 1-15 for each Round shown in [Table 4.21](#).
17. If the IUT supports the LE Coded PHY, for each round from 1 to 6 and for round 9 based on [Table 4.21](#), repeat steps 1-16, except that in Step 1 the Scanning_PHYs shall be set to 0x04 (LE Coded) and in steps 3-4 the Lower Tester advertises using the LE Coded PHY for both primary and secondary advertising channels.
18. If the IUT supports the LE 2M PHY, for each round from 1 to 6 and for round 9 based on [Table 4.21](#), repeat steps 1-16, except that in Step 1 the Scanning_PHYs shall be set to 0x01 (LE 1M) and in steps 3-4 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel.

Round	Lower Tester Periodic Advertisements (Step 4)	HCI_LE_Periodic_Advertising_Create_Sync (Steps 4 and 6)
	Periodic Data Length	Skip
1	None	0x0000
2	31 octets	0x0000
3	191 octets	0x0000
4	382 octets	0x0000
5	Scan_Max_Data	0x0000
6	Scan_Max_Data	0x0001
7	Scan_Max_Data	0x00F9
8	Scan_Max_Data	0x01F3
9	31 octets	0x000F

Table 4.21: PDU payload contents and Skip value for each case variation.

- Expected Outcome

Pass Verdict

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

- The IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements using the Channel Selection Algorithm #2.
- The IUT generates one or more HCI_LE_Periodic_Advertising_Report events. If AdvData was included in the advertisement, the Upper Tester expects to receive the correct data in the Data field(s), with Data_Status and Data_Length set correctly. The data may be fragmented across multiple events.
- The IUT maintains synchronization with the Lower Tester periodic advertisement PDUs even after the Lower Tester ceases extended advertising.
- The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.
- Where the periodic advertising data contains a counter, the difference between the counter value in consecutive reports is less than or equal to Skip + 1 in at least 95% (rounded down) of the cases.



4.2.3.19 LL/DDI/SCN/BV-23-C [Extended Scanning, Multiple Sets, Passive, Multiple PHYs (All Supported PHYs)]

- Test Purpose

Tests that a scanner IUT detects and reports advertising packets received on all supported PHYs correctly. Using all PHYs, the Lower Tester advertises using non-connectable and non-scannable extended advertising events on one channel at a time and expects the IUT to report the advertising to the Upper Tester.

- Reference

[10] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- Test Procedure

Execute the test procedure using the minimum advertising interval and continuous passive scanning on all supported PHYs.



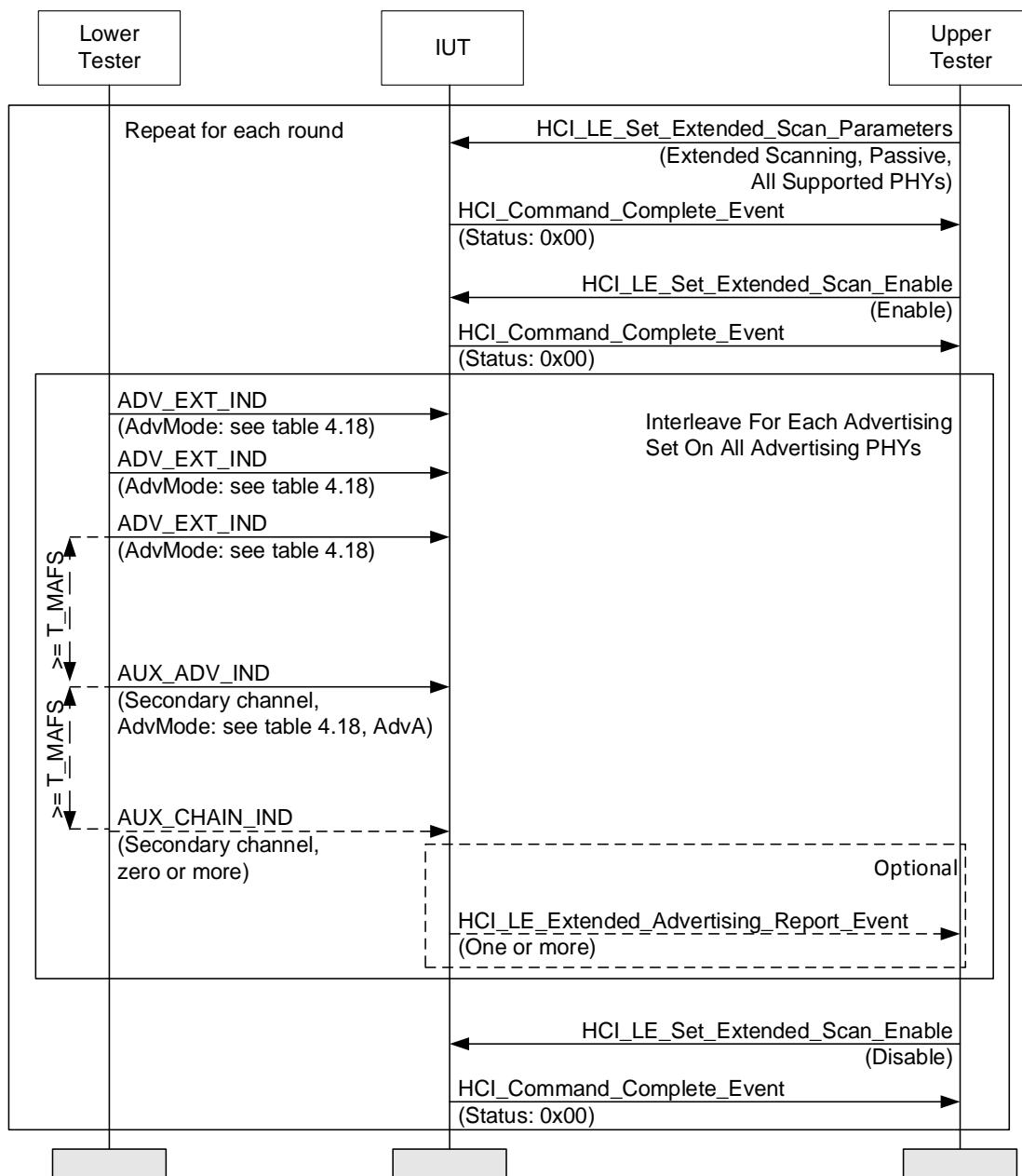


Figure 4.93: LL/DDI/SCN/BV-23-C [Extended Scanning, Multiple Sets, Passive, Multiple PHYs (All Supported PHYs)]

For each round in Table 4.22:

1. The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Parameters** Command to the IUT. The Scanning_PHYs parameter shall be set to all supported PHYs. For each supported PHY i, Scan_Type[i] shall be set to 0x00 (Passive Scanning), Scan_Interval[i] set to $(0x0010 * \text{number of bits set in Scanning_PHY})$, and Scan_Window[i] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).
2. The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Enable** command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising six advertising sets; the advertising_set ID shall be different for each set. The first two sets shall use the LE 1M PHY for both primary and secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary and

secondary advertising channels, and the last two sets shall use the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel. Each set shall use ADV_EXT_IND PDUs as specified in [Table 4.22](#) for this round with the AuxPtr field referencing the AUX_ADV_IND with all fields specified included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where "LT" equals the Lower Tester address. If AdvData is specified the PDU shall include the field populated with random octets of the specified count. If the AdvData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data. Repeat for at least 20 advertising intervals or until step 4 occurs.

4. For each set of advertisements using a PHY the IUT supports, the Upper Tester expects one or more HCI_Extended_Advertising_Report events from the IUT with an advertising event type matching the type sent in Step 3 and the Primary_PHY and Secondary_PHY set to the PHYs used for the advertising set. If AdvData was included in the advertisement, the Upper Tester expects to receive the data included in one or more of the advertising packets.
5. Upper Tester sends an HCI_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.

Round	Advertising Sets	Lower Tester Advertising PDUs		
		PDU Type (AdvMode)	AdvA	AdvData
1	1 & 2 (LE 1M, LE 1M)	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	- 191
	3 & 4 (LE Coded, LE Coded)	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	- 191
	5 & 6 (LE 1M, LE 2M)	ADV_EXT_IND (00b) AUX_ADV_IND (00b)	- LT	- 191
2	1 & 2 (LE 1M, LE 1M)	ADV_EXT_IND (00b) AUX_ADV_IND (00b) AUX_CHAIN_IND (00b)	- LT	- Scan_Max_Data
	3 & 4 (LE Coded, LE Coded)	ADV_EXT_IND (00b) AUX_ADV_IND (00b) AUX_CHAIN_IND (00b)	- LT -	- Scan_Max_Data
	5 & 6 (LE 1M, LE 2M)	ADV_EXT_IND (00b) AUX_ADV_IND (00b) AUX_CHAIN_IND (00b)	- LT	- Scan_Max_Data

Table 4.22: PDU payload contents for each case variation

- Expected Outcome

Pass Verdict

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

- For advertisements using PHYs the IUT does not support, the IUT does NOT generate an HCI_Extended_Advertising_Report event or receive the data.
- For advertisements using a PHY the IUT supports, the IUT generates one or more HCI_Extended_Advertising_Report events with an advertising event type and advertising set ID matching the type and set ID sent in Step 3 and the Primary_PHY and Secondary_PHY set to



the PHYs used for the advertising set. The Upper Tester expects to receive the correct data in the Data field, with Data_Length set correctly.

4.2.3.20 LL/DDI/SCN/BV-24-C [Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)]

- **Test Purpose**

Tests that a scanner IUT detects and requests additional information from advertising packets received on all supported PHYs, where the scan requests are sent on the secondary channel, and reports the results from the Controller. Using all PHYs, the Lower Tester advertises using scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel and expects the IUT to report the advertising to the Upper Tester.

- **Reference**

[10] 4.4.3.2

- **Initial Condition**

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (Active, selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- **Test Procedure**

Execute the test procedure using the minimum advertising interval and continuous active scanning on all supported PHYs.



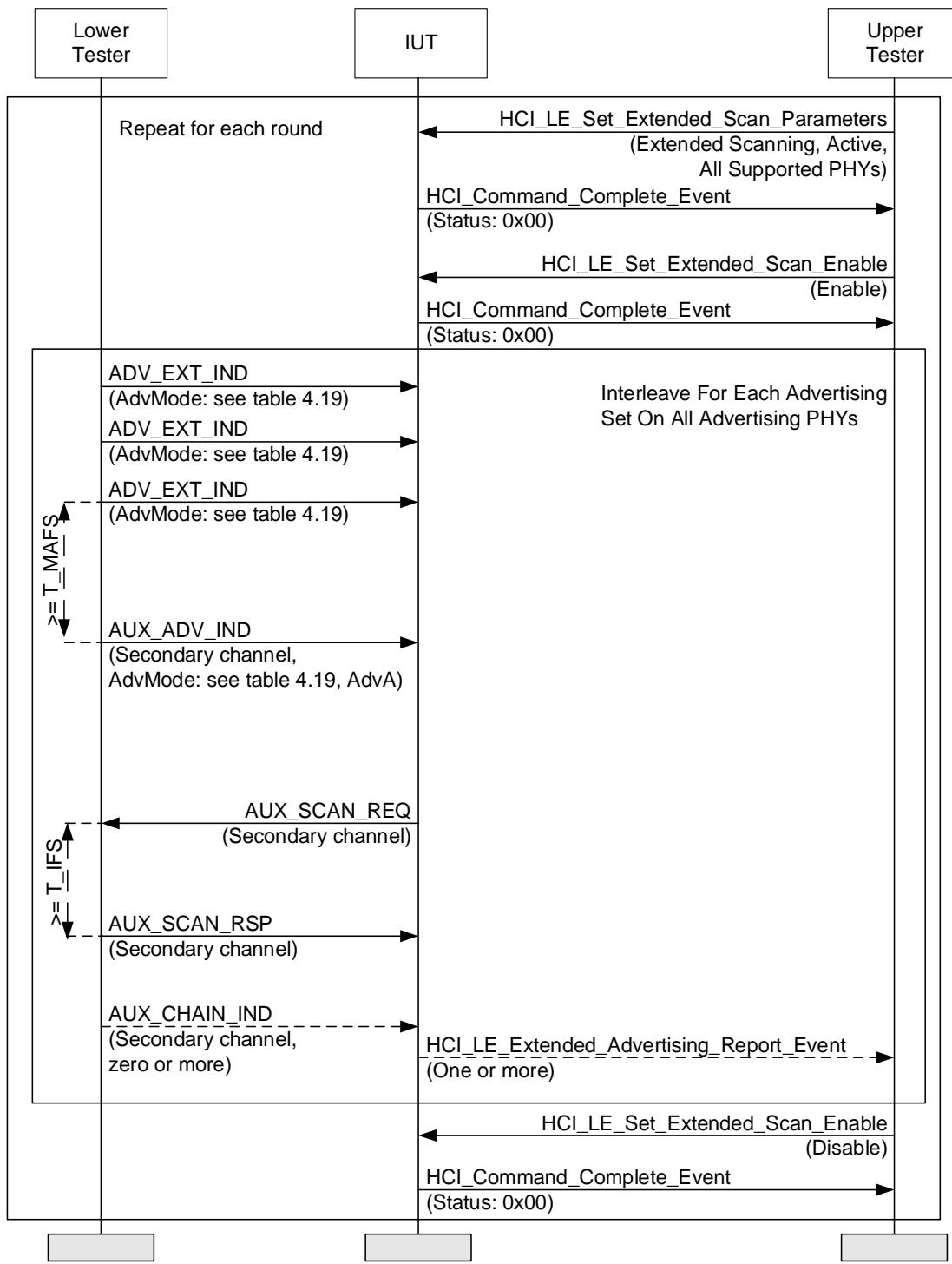


Figure 4.94: LL/DDI/SCN/BV-24-C [Extended Scanning, Multiple Sets, Active, Multiple PHYs (All Supported PHYs)]

For each round in [Table 4.23](#):

1. The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Parameters_Command** to the IUT. The **Scanning_PHYs** parameter shall be set to all supported PHYs. For each supported PHY i, **Scan_Type[i]** shall be set to 0x01 (Active Scanning), **Scan_Interval[i]** set to $(0x0010 * \text{number of bits set in Scanning_PHY})$, and **Scan_Window[i]** set to 0x0010. **Own_Address_Type** shall be set to 0x00 (Public Device Address), and **Scanning_Filter_Policy** shall be set to 0x00 (Accept All).



2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising six advertising sets; the advertising_set ID shall be different for each set. The first two sets shall use the LE 1M PHY for both primary and secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary and secondary advertising channels, and the last two sets shall use the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel. Each set shall use ADV_EXT_IND PDUs as specified in [Table 4.23](#) for this round with the AuxPtr field referencing the AUX_ADV_IND with all fields specified included with the AUX_ADV_IND only. If AdvA is specified the appropriate PDU shall include the field, where "LT" equals the Lower Tester address. Repeat for at least 20 advertising intervals or until step 4 occurs.
4. For each set of advertisements using a PHY the IUT supports, the Lower Tester expects to receive an AUX_SCAN_REQ on the appropriate advertising channel using the appropriate PHY. The ScanA field shall be set to the IUT's address and the AdvA address set to the Lower Tester's address.
5. The Lower Tester responds to each AUX_SCAN_REQ PDU with an AUX_SCAN_RSP packet to the IUT TIFS after the end of the AUX_SCAN_REQ PDU with an AdvMode of 00b. The AUX_SCAN_RSP PDU shall include the AdvMode field populated with random octets from 1 to 255, of the specified count. If the ScanData is greater in length than will fit in one PDU, the Lower Tester shall include an AuxPtr field and send one or more AUX_CHAIN_IND PDUs containing the remaining data.
6. For each scan response the Lower Tester sent, the Upper Tester expects one or more HCI_LE_Extended_Advertising_Report events from the IUT with an Event_Type where bit 3 (Scan response) is set. If ScanData was included in the response, the Upper Tester expects to receive the data included in one or more of the advertising packets.
7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.

Round	Advertising Sets	Lower Tester Advertising PDUs		
		PDU Type (AdvMode)	AdvA	ScanRspData
1	1 & 2 (LE 1M, LE 1M)	ADV_EXT_IND (10b) AUX_ADV_IND (10b)	- LT	- 191
	3 & 4 (LE Coded, LE Coded)	ADV_EXT_IND (10b) AUX_ADV_IND (10b)	- LT	- 191
	5 & 6 (LE 1M, LE 2M)	ADV_EXT_IND (10b) AUX_ADV_IND (10b)	- LT	- 191
2	1 & 2 (LE 1M, LE 1M)	ADV_EXT_IND (10b) AUX_ADV_IND (10b) AUX_CHAIN_IND (00b)	- LT	- Scan_Max_Data
	3 & 4 (LE Coded, LE Coded)	ADV_EXT_IND (10b) AUX_ADV_IND (10b) AUX_CHAIN_IND (00b)	- LT -	- Scan_Max_Data
	5 & 6 (LE 1M, LE 2M)	ADV_EXT_IND (10b) AUX_ADV_IND (10b) AUX_CHAIN_IND (00b)	- LT	- Scan_Max_Data

Table 4.23: PDU payload contents for each case variation.



- Expected Outcome

Pass Verdict

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

- For advertisements using PHYs the IUT does not support, the IUT does NOT generate an HCI_LE_Extended_Advertising_Report event or send a scan request.
- For advertisements using a PHY the IUT supports, for each set using that PHY, the IUT sends a scan request PDU to the LT.
- For scan responses received by the IUT, the IUT generates one or more HCI_LE_Extended_Advertising_Report events to the Upper Tester with an advertising event type and advertising set ID matching the type and set ID sent in Step 3. The Upper Tester expects to receive the correct data in the Data field, with Data_Length set correctly.

Inconclusive Verdict

- In the HCI_LE_Extended_Advertising_Report event received by the Upper Tester for scan responses, the Advertising SID value in the event is 0xFF.

4.2.3.21 LL/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)]

- Test Purpose

Tests that a scanner IUT can locate and receive periodic advertising events on all supported PHYs and reports the results from the Controller. Using all PHYs, the Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising and expects the IUT to report the periodic advertising data to the Upper Tester.

- Reference

[10] 4.4.2.13.1, 4.4.3.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.

- Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester using all supported PHYs.



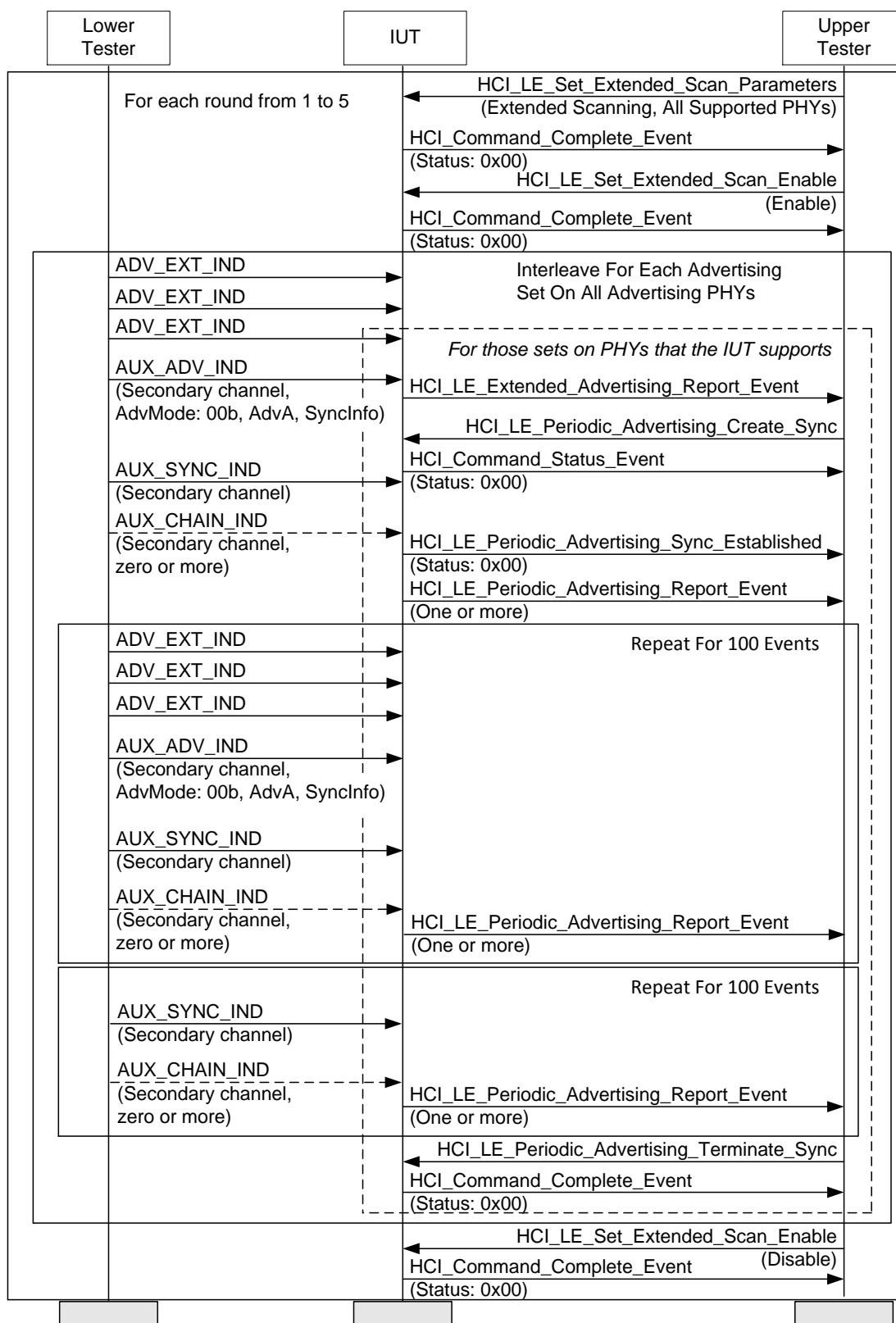


Figure 4.95: LL/DDI/SCN/BV-25-C [Extended Scanning, Multiple Sets, Periodic Advertising Reception, Multiple PHYs (All Supported PHYs)]



For each round from 1 to 5 based on [Table 4.24](#):

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters_Command to the IUT. The Scanning_PHYs parameter shall be set to all supported PHYs. For each supported PHY i, Scan_Type[i] shall be set to 0x00 (Passive Scanning), Scan_Interval[i] set to (0x0010 * number of bits set in Scanning_PHY), and Scan_Window[i] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising six advertising sets using ADV_EXT_IND and AUX_ADV_IND PDUs. The first two sets shall use the LE 1M PHY for both primary and secondary advertising channels, the next two sets shall use the LE Coded PHY for both primary and secondary advertising channels, and the last two sets shall use the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel. For each advertising set, the ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SNDC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
4. For each advertising set, the Lower Tester generates ADV_SYNC_IND PDUs on the secondary advertising channel as specified in the SyncInfo in Step 3. If the Data Length column in [Table 4.24](#) for this Round is non-zero, the AUX_SYNC_IND PDUs shall contain an AdvData field containing random octets from 1-255, up to Data Length in [Table 4.24](#) or the maximum that can fit in the PDU. If all the data cannot fit in a single AUX_SYNC_IND PDU, the Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs containing the remaining data. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
5. For each set using a PHY the IUT supports, the Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval. The first time that such an event is received for each set, steps 6 to 12 shall be performed for that set while continuing to wait for other such events. Only one instance of step 6 shall be carried out at a time but multiple instances of steps 7 to 12 may be performed in parallel with each other and with the one instance of step 6.
6.
 - a. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Filter_Policy shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from the event in Step 5. Unused shall be set to 0x00.
 - b. The Upper Tester expects an HCI_Command_Status event in response.
 - c. The Upper Tester shall wait for an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 - d. If either the HCI_Command_Status event or the HCI_LE_Periodic_Advertising_Sync_Established event indicates that the command failed because the IUT has a limitation on the number of periodic advertisements it can synchronize to at one time, this step shall be repeated when step 12 completes for another advertising set.



7. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. If AdvData is being advertised, the Upper Tester expects to receive the data included in the advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to 0xFF.
8. Repeat Step 7 until at least 100 events have been received.
9. The Lower Tester ceases extended advertising on this advertising set but continues periodic advertising.
10. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report events from the IUT as in Step 7.
11. Repeat Step 10 until at least 100 events have been received.
12. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
13. When step 12 has completed for all sets using a PHY the IUT supports, the Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
14. The Lower Tester ceases periodic advertising on all advertising sets.
15. Repeat steps 1-14 for each Round shown in [Table 4.24](#).

Round	Lower Tester Periodic Advertisements (Step 4)
	Periodic Data Length
1	None
2	31 octets
3	191 octets
4	382 octets
5	Scan_Max_Data

Table 4.24: PDU payload contents for each case variation.

- Expected Outcome

Pass Verdict

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

- For each set using a PHY the IUT supports, the IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements.
- For each set using a PHY the IUT supports, the IUT generates one or more HCI_LE_Periodic_Advertising_Report events. If AdvData was included in the advertisement, the Upper Tester expects to receive the correct data in the Data field(s), with Data_Status and Data_Length set correctly. The data may be fragmented across multiple events.
- For each set using a PHY the IUT supports, the IUT maintains synchronization with the Lower Tester periodic advertisement PDUs even after the Lower Tester ceases extended advertising.
- For each set using a PHY the IUT supports, the IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.



- For PHYs the IUT does not support, the IUT does not generate HCI_LE_Periodic_Advertising_Report events.

4.2.3.22 LL/DDI/SCN/BV-26-C [Network Privacy – Passive Scanning, Peer IRK, Ignore Identity Address]

- Test Purpose

Verify the IUT when doing passive scanning and using the Resolving List does not report advertising from the Lower Tester using its device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode. The Lower Tester has distributed an IRK, but the Lower Tester is doing non-connectable advertising using its device identity address for the AdvA field.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertisement packets (0x00)).

The IUT is using a resolvable private address (0x02 or 0x03).

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.



- Test Procedure

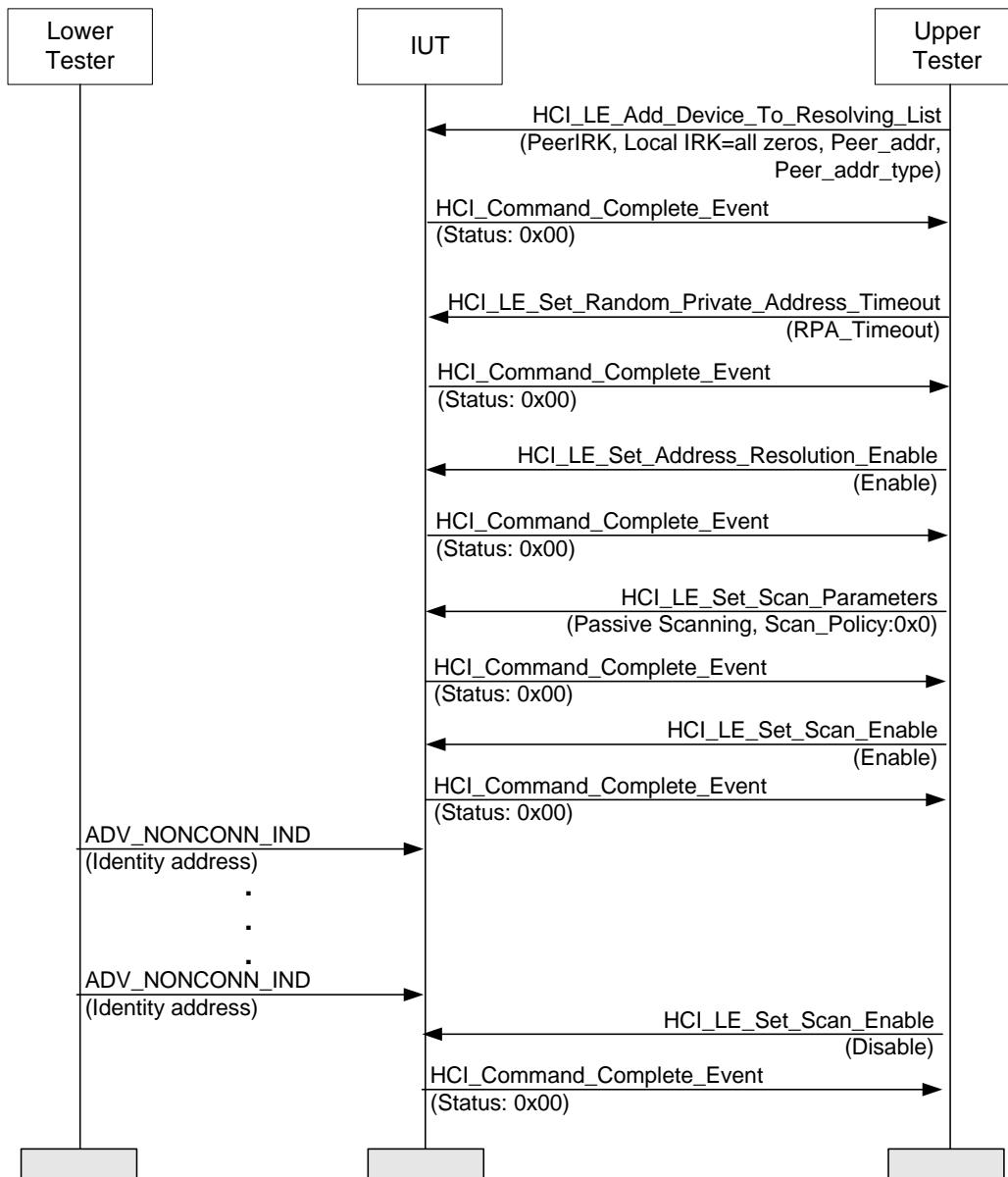


Figure 4.96: LL/DDI/SCN/BV-26-C [Network Privacy – Passive Scanning, Peer IRK, Ignore Identity Address]

- The Upper Tester populates the IUT resolving list with the peer IRK and identity address.
- The Upper Tester enables passive scanning in the IUT.
- Configure the Lower Tester to start advertising. The Lower Tester uses its device identity address in the AdvA field.
- The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
- The Upper Tester expects to receive no HCI_LE_Advertising_Report events from the IUT.
- The Upper Tester sends an HCI_Command_Complete_Event to the IUT to stop the scanning function and expects an HCI_Command_Complete_event in response.

- Expected Outcome

Pass Verdict

The IUT does not report advertising with the device identity address in the AdvA field from the Lower Tester, when the device identity address and an associated IRK are in the resolving list using network privacy mode.

4.2.3.23 LL/DDI/SCN/BV-28-C [Device Privacy – Passive Scanning, Peer IRK, Accept Identity Address]

- Test Purpose

Verify the IUT when doing passive scanning and using the Resolving List reports advertising from the Lower Tester using its device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode. The Lower Tester has distributed an IRK, but the Lower Tester is doing non-connectable advertising using its device identity address for the AdvA field.

- Reference

[3] 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND if filtering is supported filtering policy to allow all devices (accept all advertisement packets (0x00)).



- Test Procedure

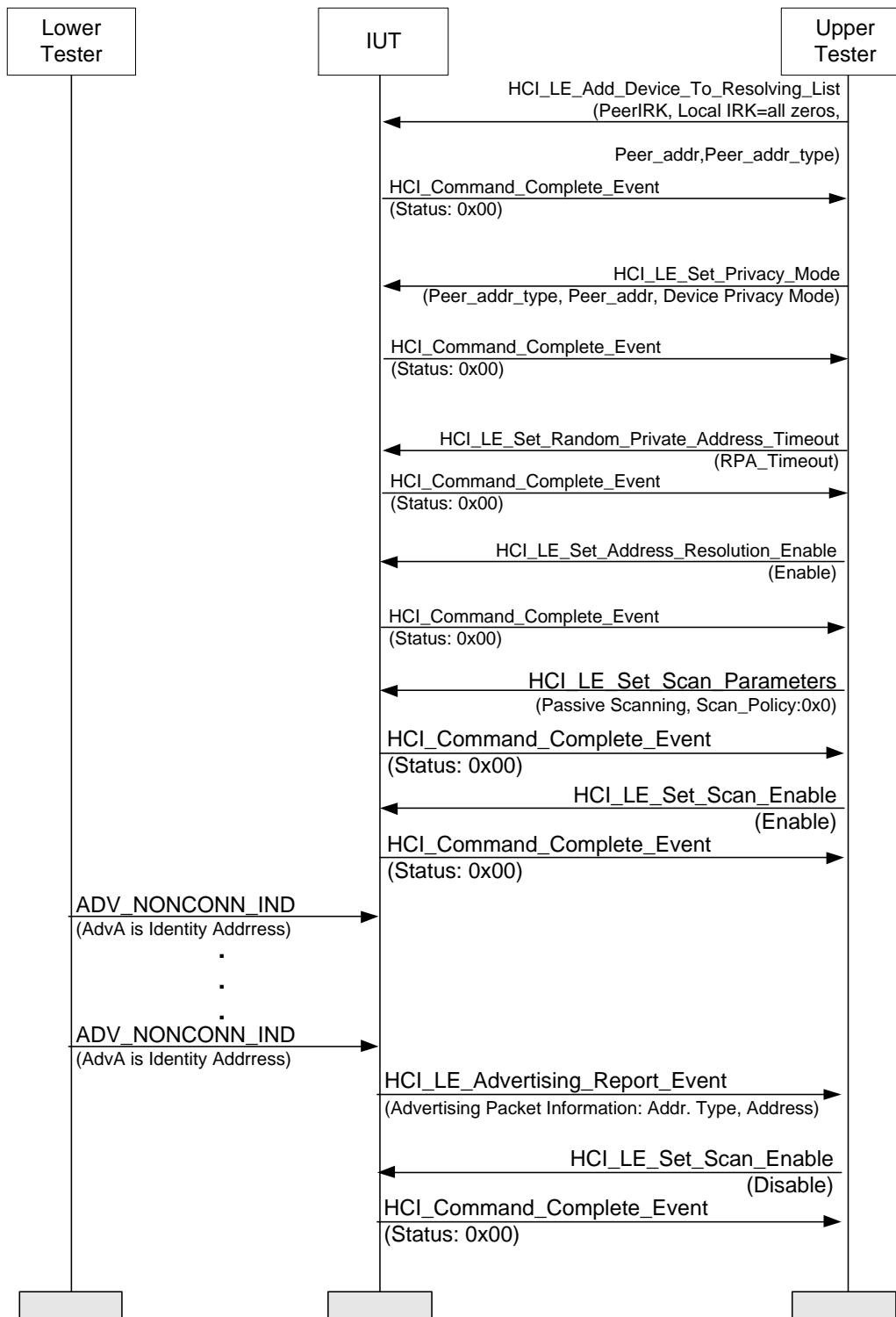


Figure 4.97: LL/DDI/SCN/BV-28-C [Device Privacy – Passive Scanning, Peer IRK, Accept Identity Address]

- The Upper Tester populates the IUT resolving list with the peer IRK and identity address and sets the entry to device privacy mode.
- The Upper Tester enables passive scanning in the IUT.



3. Configure the Lower Tester to start advertising. The Lower Tester uses its device identity address in the AdvA field.
 4. The Lower Tester sends an ADV_NONCONN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals.
 5. The Upper Tester expects at least one HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester. The address in the report is the Lower Tester's identity address.
 6. The Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to stop the scanning function and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The IUT receives and reports advertising with the device identity address in the AdvA field from the Lower Tester, when the device identity address and an associated IRK are in the resolving list using device privacy mode.

4.2.3.24 LL/DDI/SCN/BV-29-C [AoD Connectionless CTE Scanning]

- Test Purpose

Tests that a scanner IUT detects and reports advertising events with the AoD Connectionless Constant Tone Extension included when advertiser is utilizing a public device address.

The Lower Tester is advertising with an AoD Connectionless Constant Tone Extension, using a public device address. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- Reference

[\[13\]](#) 2.5.2, 2.5.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).



- Test Procedure

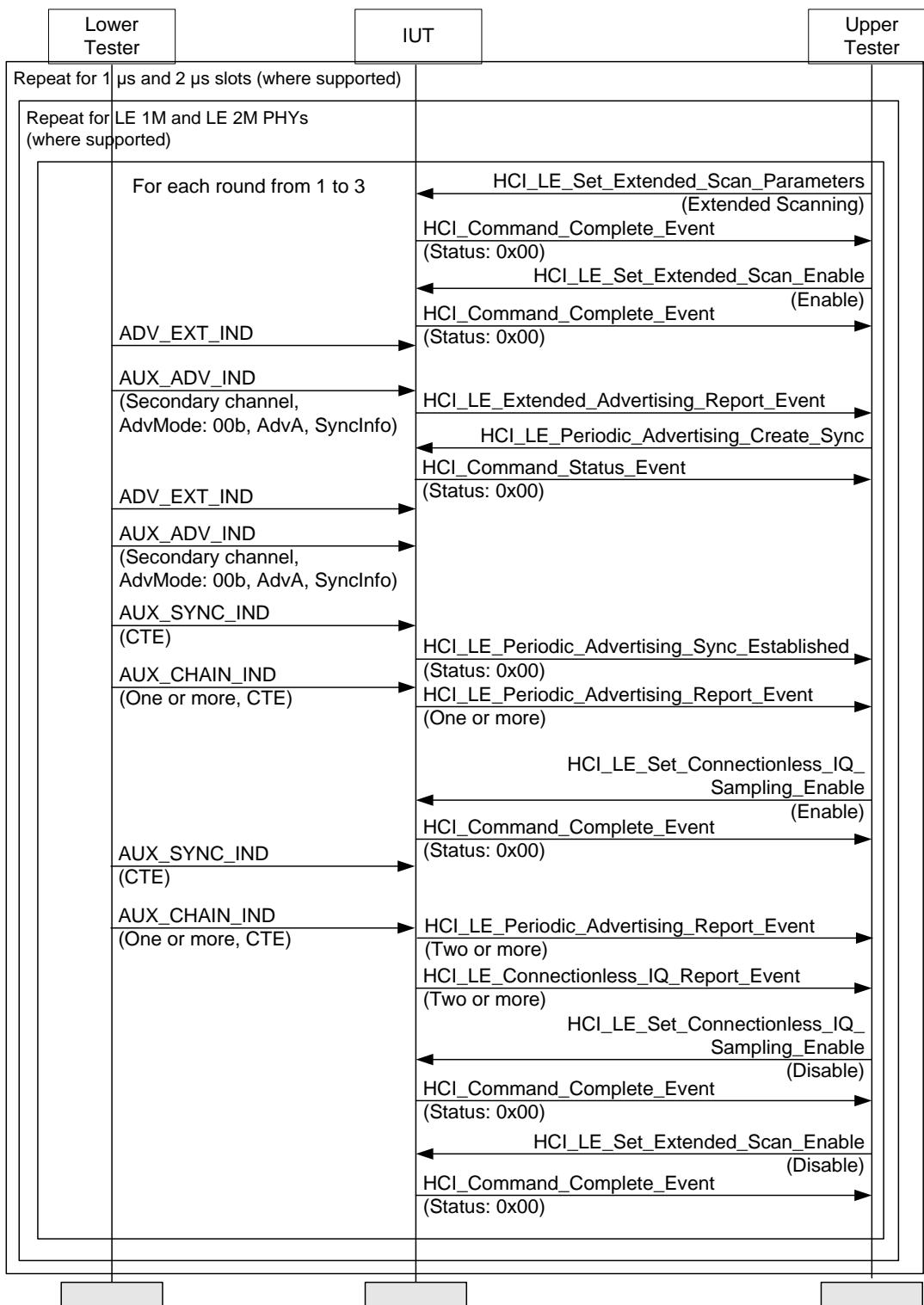


Figure 4.98: LL/DDI/SCN/BV-29-C [AoD Connectionless CTE Scanning]



For each round from 1 to 3 based on [Table 4.25](#):

Round	CTETime (Step 4)
1	0x02
2	0x0A
3	0x14

Table 4.25: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the Adva field containing the Lower Tester address, a SyncInfo field referring to the AUX_SYNC_IND PDUs and does not include the AdvData field. The Lower Tester continues advertising until directed to stop in the test procedure.
4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to the value specified in [Table 4.25](#), RFU set to '0', and CTEType set to 2 (AoD Constant Tone Extension with 2 μ s slots). The AUX_SYNC_IND PDUs shall not include the AdvData field. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in [Table 4.25](#). While transmitting the Constant Tone Extension field, the Lower Tester switches antennae using 2 μ s slots. The Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs.
5. The Lower Tester generates one or more AUX_CHAIN_IND PDUs on the secondary advertising channel. The AUX_CHAIN_IND PDUs shall not include the AdvData field. Each packet containing an AUX_CHAIN_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in [Table 4.25](#). While transmitting the Constant Tone Extension field, the Lower Tester switches antennae using 2 μ s slots. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
6. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.
7. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from step 6. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 6. CTE_Type shall be set to 0x13 (accept only AoD Constant Tone Extension with 2 μ s slots). The Upper Tester expects an HCI_Command_Status event in response.



8. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 9. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The RSSI shall be set to a value in the range of -127 – 20 dBm. CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots).
 10. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT. Sampling_Enable shall be set to 0x01 (enabled). Max_Sampled_CTEs shall be set to 0x00 (Sample and report all available Constant Tone Extensions). Sync_Handle shall be set to the Sync_Handle from step 8. The Upper Tester expects an HCI_Command_Complete event from the IUT with Status set to 0x00 (Success) and Sync_Handle set to the Sync_Handle from step 8.
 11. The Upper Tester expects two or more HCI_LE_Connectionless_IQ_Report events from the IUT. The Sync_Handle shall be set to a value in the range of 0x0000 - 0x0EFF. The Channel_Index shall be set to a value in the range of 0x00 - 0x24. The Packet_Status shall be set to 0x00 (CRC was correct). The RSSI shall be set to a valid value. The CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots). The paEventCounter shall be set to the EventCouter value from the SyncInfo field of the AUX_SYNC_IND. Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 4$. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.
 12. The Lower Tester ceases extended advertising but continues periodic advertising.
 13. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report events and HCI_LE_Connectionless_IQ_Report events from the IUT as in steps 9 and 11.
 14. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT to disable sampling.
 15. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.
 16. Repeat steps 1-15 for each round shown in [Table 4.25](#).
 17. If the IUT supports the LE 2M PHY, repeat steps 1-16, except that in step 1 the Scanning_PHYs shall be set to 0x01 (LE 1M) and that in steps 3-5 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel.
 18. If the IUT supports 1 μ s slots, repeat steps 1-17, except that in step 4 the CTE_Type shall be set to 1 (AoD Constant Tone Extension with 1 μ s slots) and the Lower Tester switches antenna using 1 μ s slots, that in step 7 CTE_Type shall be set to 0x15 (accept only AoD Constant Tone Extension with 1 μ s slots), and that in step 11 the CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 μ s slots) and the Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 2$.
- Expected Outcome
- Pass Verdict
- For all rounds described in the test procedure, the following condition shall occur:
- The IUT generates two or more HCI_LE_Connectionless_IQ_Report events. HCI_LE_Connectionless_IQ_Report events shall be sent for packets received with AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs.
 - The IUT maintains synchronization with the Lower Tester's periodic advertisement PDUs and continues IQ sampling even after the Lower Tester ceases extended advertising.
 - The IUT terminates periodic advertising reception and IQ sampling when the Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command.



4.2.3.25 LL/DDI/SCN/BV-30-C [AoA Connectionless CTE Scanning]

- Test Purpose

Tests that a scanner IUT detects and reports advertising events with the AoA Connectionless Constant Tone Extension included when advertiser is utilizing a public device address.

The Lower Tester is advertising with an AoA Connectionless Constant Tone Extension, using a public device address. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- Reference

[13] 2.5.2, 2.5.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.



- Test Procedure

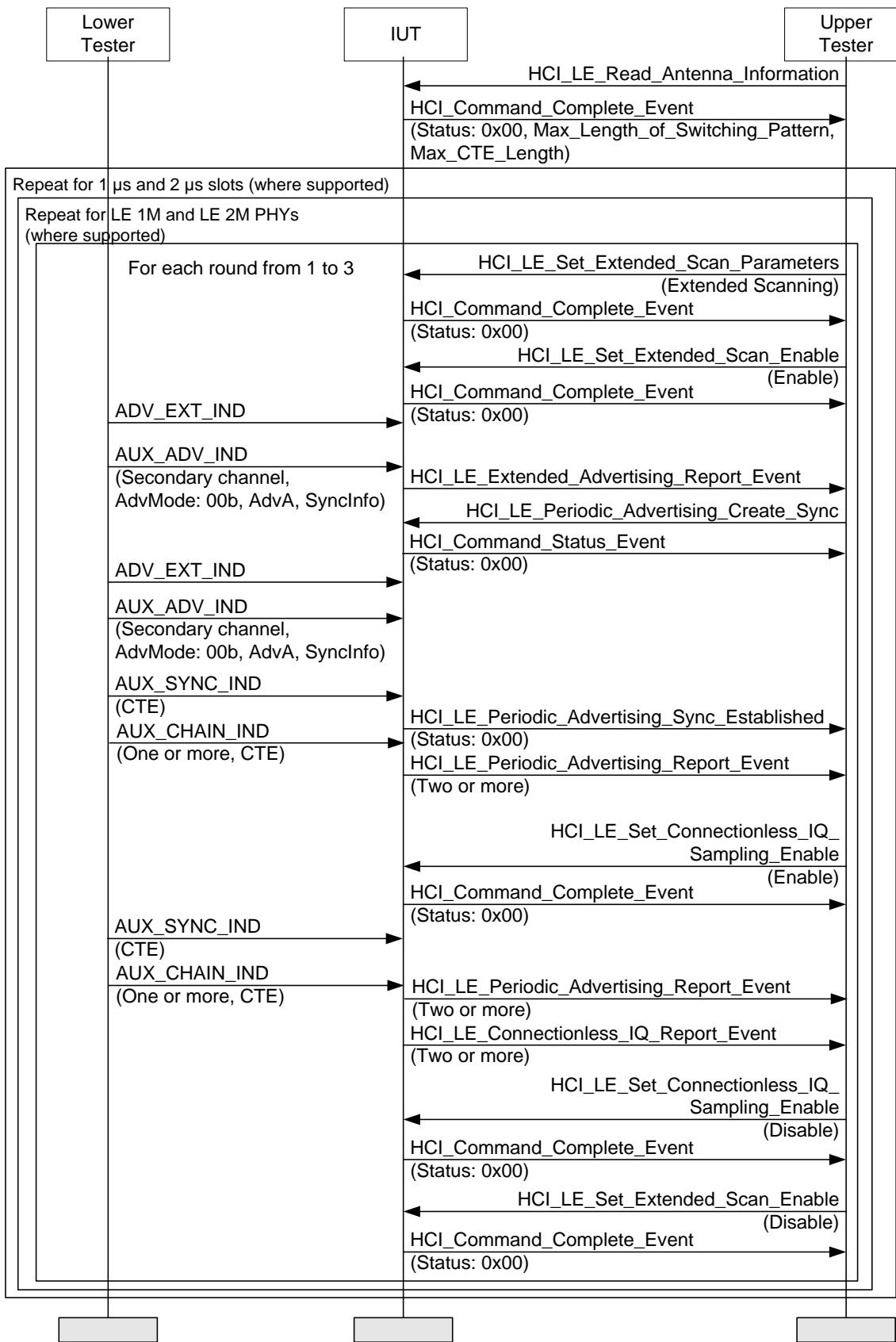


Figure 4.99: LL/DDI/SCN/BV-30-C [AoA Connectionless CTE Scanning]



- The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_Length_Switching_Pattern between 0x02 and 0x4B. The Upper Tester stores the Max_Length_Switching_Pattern for future use.

For each round from 1 to 3 based on [Table 4.26](#):

Round	CTETime (Step 5)
1	0x02
2	0x0A
3	0x14

Table 4.26: Parameter values for each case variation

- The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
- The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
- The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address, a SyncInfo field referring to the AUX_SYNC_IND PDUs and does not include the AdvData field. The Lower Tester continues advertising until directed to stop in the test procedure.
- The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to the value specified in [Table 4.26](#), RFU set to '0', and CTEType set to 0 (AoA Constant Tone Extension). The AUX_SYNC_IND PDUs shall not include the AdvData field. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in [Table 4.26](#). The Lower Tester shall include an AuxPtr in the AUX_SYNC_IND PDU which refers to one or more AUX_CHAIN_IND PDUs.
- The Lower Tester generates one or more AUX_CHAIN_IND PDUs on the secondary advertising channel. The AUX_CHAIN_IND PDUs shall not include the AdvData field. Each packet containing an AUX_CHAIN_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime specified in [Table 4.26](#). While transmitting the Constant Tone Extension field, the Lower Tester switches antennae using 2 μ s slots. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
- The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.
- The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 7. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to 3 x Periodic_Advertising_Interval from step 7. CTE_Type shall be set to 0x16 (accept only AoA Constant Tone Extension). The Upper Tester expects an HCI_Command_Status event in response.



9. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 10. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The RSSI shall be set to a value in the range of -127 – 20 dBm. CTE_Type shall be set to 0x00 (AoA Constant Tone Extension).
 11. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT. Sampling_Enable shall be set to 0x01 (enabled). Max_Sampled_CTEs shall be set to 0x00 (Sample and report all available Constant Tone Extensions). Sync_Handle shall be set to the Sync_Handle from step 9. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. Slot_Durations shall be set to 0x02 (2 μ s slots). The Upper Tester expects an HCI_Command_Complete event from the IUT with Status set to 0x00 (Success) and Sync_Handle set to the Sync_Handle from step 9.
 12. The Upper Tester expects an HCI_LE_Connectionless_IQ_Report event from the IUT. The Sync_Handle shall be set to a value in the range of 0x0000 - 0x0EFF. The Channel_Index shall be set to a value in the range of 0x00 - 0x24. The Packet_Status shall be set to 0x00 (CRC was correct). The RSSI shall be set to a valid value. The RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array at step 11. The CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). Slot_Durations shall be set to 0x02 (2 μ s slots). The paEventCounter shall be set to the EventCouter value from the SyncInfo field of the AUX_SYNC_IND. Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 4$. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.
 13. The Lower Tester ceases extended advertising but continues periodic advertising.
 14. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report events and HCI_LE_Connectionless_IQ_Report events from the IUT as in steps 10 and 12.
 15. The Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command to the IUT to disable sampling.
 16. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.
 17. Repeat steps 2-16 for each round shown in [Table 4.26](#).
 18. If the IUT supports the LE 2M PHY, repeat steps 2-17, except that in step 1 the Scanning_PHYs shall be set to 0x01 (LE 1M) and that in steps 4-6 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel.
 19. If the IUT supports 1 μ s slots, repeat steps 2-18, except that in step 11 Slot_Durations shall be set to 0x01 (1 μ s slots) and that in step 12 Slot_Durations shall be set to 0x01 (1 μ s slots) and the Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 2$.
- Expected Outcome
- Pass Verdict
- For all rounds described in the test procedure, the following condition shall occur:
- The IUT generates two or more HCI_LE_Connectionless_IQ_Report events. HCI_LE_Connectionless_IQ_Report events shall be sent for packets received with AUX_SYNC_IND PDUs or AUX_CHAIN_IND PDUs.
 - The IUT maintains synchronization with the Lower Tester's periodic advertisement PDUs and continues IQ sampling even after the Lower Tester ceases extended advertising.



- The IUT terminates periodic advertising reception and IQ sampling when the Upper Tester sends an HCI_LE_Set_Connectionless_IQ_Sampling_Enable command.

4.2.3.26 LL/DDI/SCN/BV-31-C [AoD Connectionless CTE Scanning, Incorrect CRC]

- Test Purpose

Tests that a scanner IUT detects and reports advertising events with the AoD Connectionless Constant Tone Extension included when receiving packets with incorrect CRCs.

The Lower Tester is advertising with an AoD Connectionless Constant Tone Extension, using incorrect CRCs. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- Reference

[13] 2.5.2, 2.5.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

- Test Procedure

As for LL/DDI/SCN/BV-29-C (AoD Connectionless CTE Scanning), except that in step 4, the Lower Tester transmits the packet without the Constant Tone Extension, and in step 5 the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC.

The Upper Tester expects to receive only one HCI_LE_Periodic_Advertising_Report event from the IUT in step 9, and only one or more HCI_LE_Connectionless_IQ_Report events with Packet_Status set to 0x01 or 0x02 from the IUT in step 11.

- Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connectionless_IQ_Report events with Packet_Status set to indicate the CRC was incorrect.

Inconclusive Verdict

The IUT does not generate any HCI_LE_Connectionless_IQ_Report events.

4.2.3.27 LL/DDI/SCN/BV-32-C [AoA Connectionless CTE Scanning, Incorrect CRC]

- Test Purpose

Tests that a scanner IUT detects and reports advertising events with the AoA Connectionless Constant Tone Extension included when receiving packets with incorrect CRCs.

The Lower Tester is advertising with an AoA Connectionless Constant Tone Extension, using incorrect CRCs. The Upper Tester configures the IUT to scan for Connectionless Constant Tone Extension from the Lower Tester. The Upper Tester observes the IQ report events generated.

- Reference

[13] 2.5.2, 2.5.4



- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- Test Procedure

As for LL/DDI/SCN/BV-30-C (AoA Connectionless CTE Scanning), except that in step 4 the Lower Tester transmits the packet without the Constant Tone Extension, and in step 5 the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC.

The Upper Tester expects to receive only one HCI_LE_Periodic_Advertising_Report event from the IUT in step 10, and only one or more HCI_LE_Connectionless_IQ_Report events with Packet_Status set to 0x01 or 0x02 from the IUT in step 12.

- Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connectionless_IQ_Report events with Packet_Status set to indicate the CRC was incorrect.

Inconclusive Verdict

The IUT does not generate any HCI_LE_Connectionless_IQ_Report events.

4.2.3.28 LL/DDI/SCN/BV-33-C [Privacy - Extended Scanning, Active]

- Test Purpose

Tests that a scanner IUT detects and reports a directed scannable advertising packet according to the scan filter policy set by the Upper Tester. The IUT also responds with a scan request and reports the scan response.

- Reference

[10] 6.3, 4.4.2.8

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (selected scan interval, selected scan window) AND All White Listed (policy for scanner).

The IUT is using a resolvable private address (0x02 or 0x03).

The Lower Tester is also using a resolvable private address in the AdvA field of the advertising packets.

The Lower Tester has previously distributed its IRK to the IUT

- Test Procedure

Execute the test procedure using the minimum advertising interval and continuous active scanning



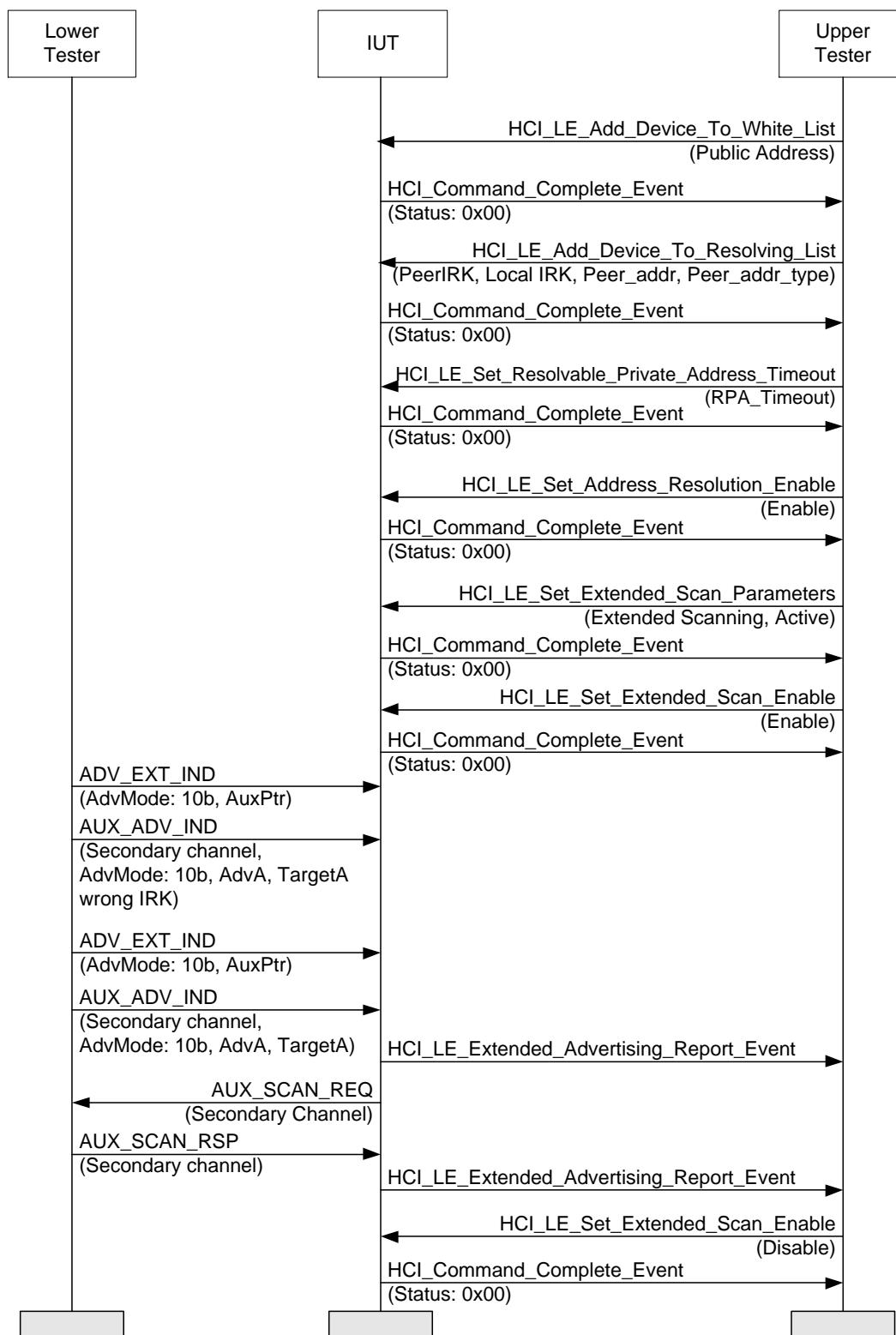


Figure 4.100: LL/DDI/SCN/BV-33-C [Network Privacy - Extended Scanning, Passive]

1. The Upper Tester adds public address of the Lower Tester to the white list.
2. The Upper Tester adds peer device identity and local IRK information to resolving list.

3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x01 (Active Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x02 or 0x03 (Resolvable Private Address), and Scanning_Filter_Policy shall be set to 0x01 (Accept only advertising packets from devices where the advertiser's address is in the White List).
4. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
5. The Lower Tester begins advertising using the ADV_EXT_IND PDU on the LE 1M PHY with AdvMode field set to 10b ("Scannable") and the AuxPtr field referencing the AUX_ADV_IND on the LE 1M PHY. The AUX_ADV_IND PDU shall include the AdvMode field set to 10b ("Scannable"), the AdvA field with a valid resolvable private address, and the TargetA field using a resolvable private address generated from a random IRK different from the one distributed to the IUT.
6. The IUT tries to resolve the address in the TargetA field by checking against its resolving list and does not find a match.
7. Repeat steps 5-6 for at least 20 advertising intervals
8. The Lower Tester begins advertising again with the TargetA field using a resolvable private address generated from the correct IRK.
9. Repeat step 8 for at least 20 advertising intervals or until step 11.
10. The Upper Tester expects one HCI_LE_Extended_Advertising_Report event from the IUT with an Event_Type where
 - bit 1 ("Scannable advertising") and bit 2 ("Directed Advertising") are set,
 - bit 0 ("Connectable advertising") and bit 4 ("Legacy advertising PDUs used") are cleared,
 - other bits may have any value.
11. The Lower Tester expects an AUX_SCAN_REQ on the appropriate advertising channel. The ScanA field shall be set to the IUT's address and the AdvA address set to the Lower Tester's address. The Upper Tester expects to receive the Direct Address Type and Direct Address used to direct the advertisement at the IUT.
12. The Lower Tester responds with an AUX_SCAN_RSP packet to the IUT T_IFS after the end of the AUX_SCAN_REQ PDU with an AdvMode of 00b.
13. The Upper Tester expects one HCI_LE_Extended_Advertising_Report event from the IUT with an Event_Type where
 - bit 1 ("Scannable advertising"), bit 2 ("Directed Advertising") and bit 3 ("Scan response") are set,
 - bit 0 ("Connectable advertising"), bit 4 ("Legacy advertising PDUs used") and bits 5-6 ("Data Status") are cleared,
 - other bits may have any value.
14. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.

- Expected Outcome

Pass Verdict

The IUT receives scannable directed advertising with a resolvable private address in the AdvA field and an invalid or valid resolvable private address in the TargetA field from the Lower Tester.



For advertisements where TargetA address does not match the resolving list, the IUT does NOT generate a scan request and an HCI_LE_Extended_Advertising_Report event.

For advertisements where TargetA address matches the resolving list, the IUT generates a scan request and an HCI_LE_Extended_Advertising_Report event.

4.2.3.29 LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]

- Test Purpose

Tests that a scanner IUT can synchronize to periodic advertising events using both periodic sync establishment filter policy settings, and ignoring the scanner filter policy.

- Reference

[10] 4.3.5, 4.4.3.4

- Initial Condition

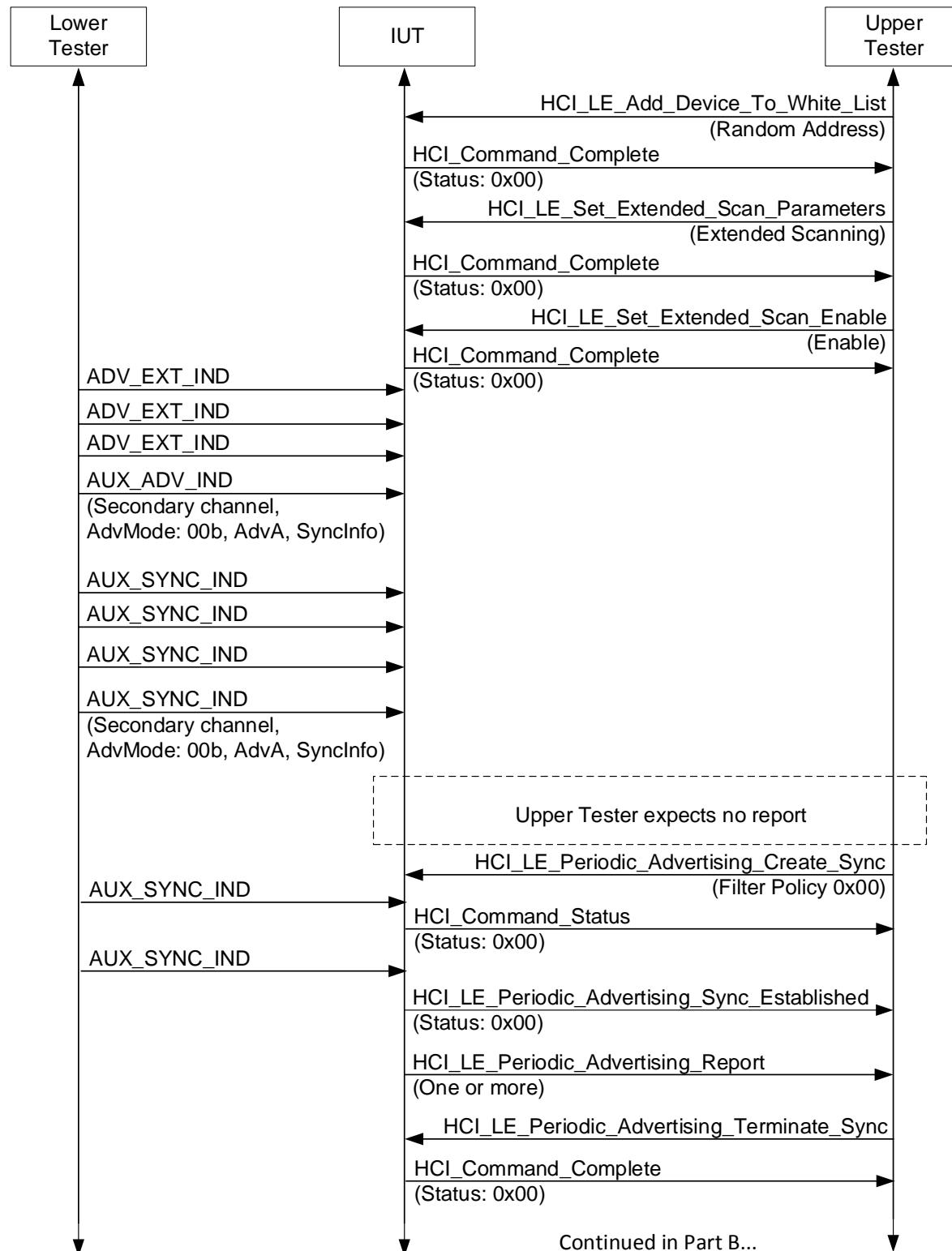
Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Idle.

The maximum number of octets the IUT can receive during scanning is defined in the “Scan Max Data” IXIT parameter.



- Test Procedure



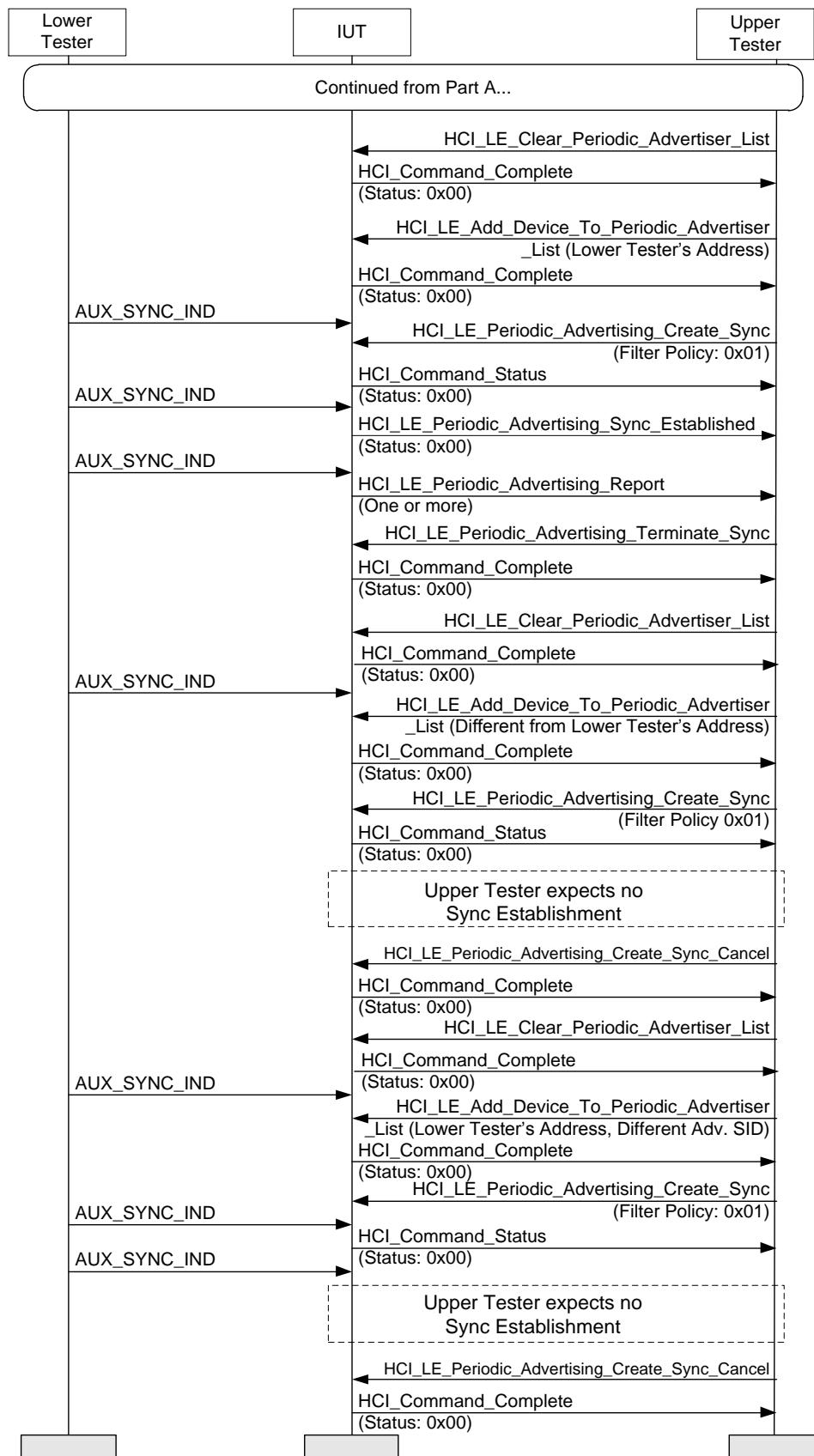


Figure 4.101: LL/DDI/SCN/BV-34-C [Extended Scanning, Periodic Advertising Reception, Filter Policies]

1. The Upper Tester sends an HCI_LE_Add_Device_To_White_List command to the IUT, containing a randomly generated device address that is different than the Lower Tester's address and expects an HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type shall be set to 0x00 (Public Device Address), and Scanning_Filter_Policy shall be set to 0x01 (use White List). The Upper Tester expects an HCI_Command_Complete event in response.
3. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero. The Upper Tester expects an HCI_Command_Complete event in response.
4. The Lower Tester begins non-connectable, non-scannable, undirected advertising using ADV_EXT_IND and AUX_ADV_IND PDUs, on the LE 1M PHY. The ADV_EXT_IND PDUs shall include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs shall include the AdvA field containing the Lower Tester's address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
5. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in step 3. The advertising data is chosen to fit in a single AUX_SYNC_IND PDU. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Filter_Policy shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the value used by the Lower Tester in the periodic advertisements. Advertiser_Address_Type and Advertiser_Address shall be set to match the Lower Tester's address. Skip shall be set to 0x0000. Sync_Timeout shall be set to three times the periodic advertising interval. Unused shall be set to 0x00. The Upper Tester expects an HCI_Command_Status event in response.
7. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
8. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. The Upper Tester expects to receive the data included in the advertising packet(s) sent to the Upper Tester in one or more events. Unused shall be set to 0xFF.
9. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
10. Upper Tester sends an HCI_LE_Clear_Periodic_Advertiser_List command, followed by an HCI_LE_Add_Device_To_Periodic_Advertiser_List command containing the Lower Tester's address and the Advertising SID used in the periodic advertisements. The Upper Tester expects an HCI_Command_Complete event in response to each command.
11. Repeat steps 6-9, but in step 6 the Filter_Policy is set to 0x01 (Use Advertiser List).
12. Repeat step 10 using a different advertiser address.
13. Repeat step 6 with the Filter_Policy set to 0x01. The Upper Tester expects the IUT not to synchronize with the periodic advertisements of the Lower Tester.
14. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command and expects an HCI_Command_Complete event in response (indicating success).
15. Repeat step 10 using the correct advertiser address but a different Advertising SID.
16. Repeat step 6 with the Filter_Policy set to 0x01. The Upper Tester expects the IUT not to synchronize with the periodic advertisements of the Lower Tester.



17. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command and expects an HCI_Command_Complete event in response (indicating success).

- Expected Outcome

Pass Verdict

After the first and second times that step 6 is executed, the IUT is able to synchronize to the periodic advertisements. After the third and fourth times, the IUT does not synchronize to the periodic advertisements.

Fail Verdict

The IUT generates extended advertising reports at any time throughout the test.

4.2.3.30 LL/DDI/SCN/BV-35-C [Connectionless CTE Scanning, Filter Wrong CTE Types on Synchronization]

- Test Purpose

Tests that a scanner IUT applies the filtering policy while attempting synchronization with the periodic advertising, when the Lower Tester uses a wrong Constant Tone Extension type of the periodic advertisements.

- Reference

[13] 2.5.2, 2.5.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).



- Test Procedure

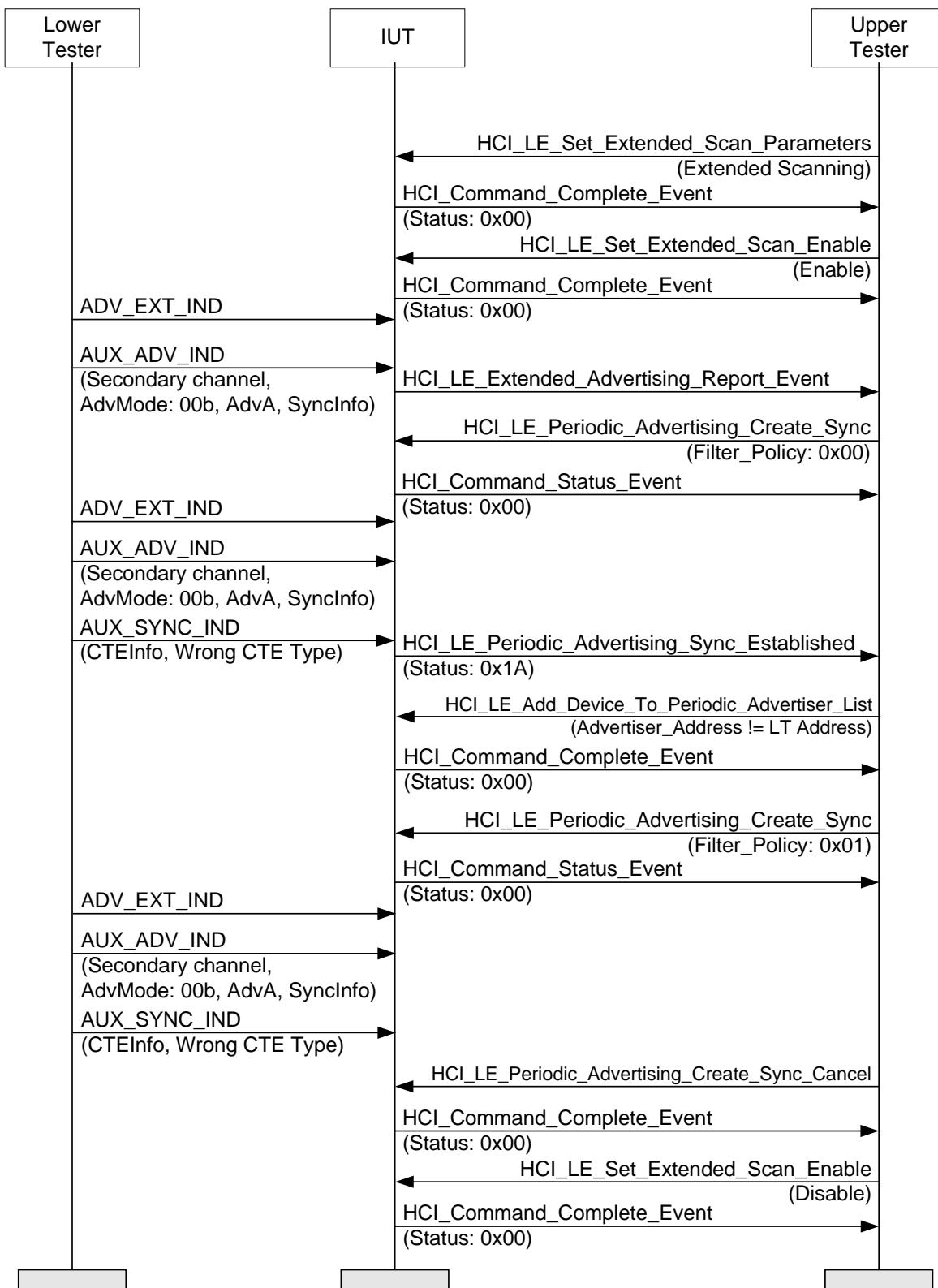


Figure 4.102: LL/DDI/SCN/BV-35-C [Connectionless CTE Scanning, Filter Wrong CTE Types on Synchronization]

1. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT. The Scanning_PHYs parameter shall be set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, Scan_Window[0] set to 0x0010, Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the Adva field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to any valid value, RFU set to '0', and CTEType set to a valid value for a CTE type that the IUT supports. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
5. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.
6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to $3 \times$ Periodic_Advertising_Interval from Step 5. CTE_Type shall be set to a value that specifies not to synchronize to packets with a CTE type from Step 4. The Upper Tester expects an HCI_Command_Status event in response.
7. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x1A (Unsupported Remote Feature / Unsupported LMP Feature) and other fields matching the advertisements generated by the Lower Tester.
8. The Upper Tester does not expect any HCI_LE_Periodic_Advertising_Report events from the IUT.
9. The Upper Tester sends an HCI_LE_Add_Device_To_Periodic_Advertiser_List command. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to a valid address other than the Lower Tester's address. Advertising_SID shall be set to a valid value other than the Advertising_SID from Step 5. The Upper Tester expects an HCI_Command_Status event in response.
10. Repeat Steps 5-6 with Options set to 0x01 (Use List).
11. The Upper Tester does not expect any HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) or HCI_LE_Periodic_Advertising_Report events from the IUT.
12. The Upper Tester send an HCI_LE_Periodic_Advertising_Create_Sync_Cancel command to the IUT to cancel the pending HCI_LE_Periodic_Advertising_Create_Sync command and expects an HCI_Command_Status event in response.
13. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.



- Expected Outcome

Pass Verdict

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

- The IUT does not synchronize with the Lower Tester's periodic advertisement PDUs.
- When filtering policy is set to 0x01, the IUT continues synchronization until the procedure is canceled by the Upper Tester.
- When filtering policy is set to 0x00, the IUT stops the synchronization procedure.

4.2.3.31 LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]

- Test Purpose

Tests that a scanner IUT maintains synchronization with the periodic advertising, when the Lower Tester changes the Constant Tone Extension type or does not transmit any Constant Tone Extension of in the periodic advertisements.

- Reference

[13] 2.5.2, 2.5.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertisement packets (0x00)).



- Test Procedure

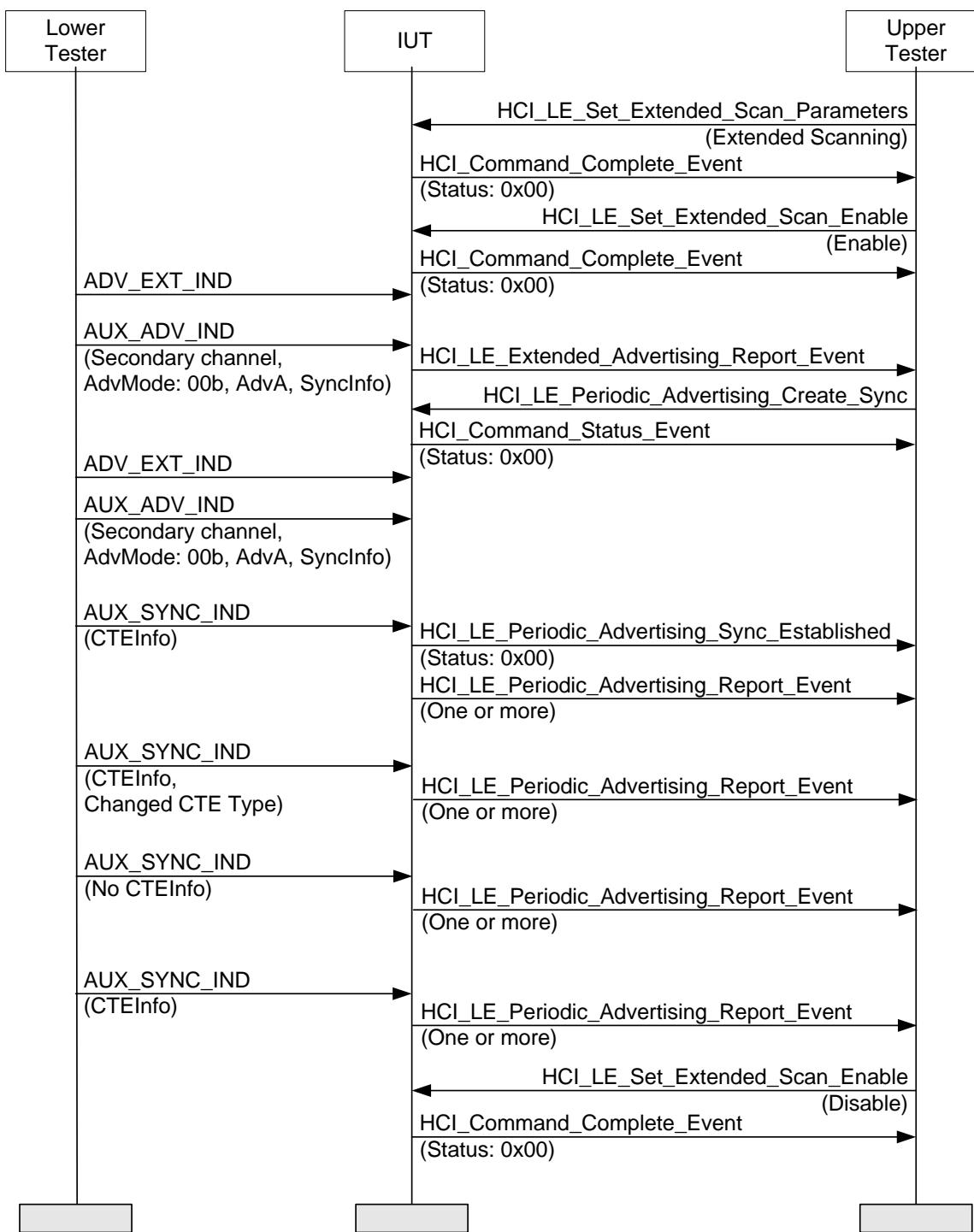


Figure 4.103: LL/DDI/SCN/BV-36-C [Connectionless CTE Scanning, CTE Type Change]

1. The Upper Tester sends an **HCI_LE_Set_Extended_Scan_Parameters** command to the IUT. The **Scanning_PHYs** parameter shall be set to 0x01 (LE 1M), **Scan_Type[0]** set to 0x00 (Passive Scanning), **Scan_Interval[0]** set to 0x0010, **Scan_Window[0]** set to 0x0010,



- Own_Address_Type[0] set to 0x00 (Public Device Address), and Scanning_Filter_Policy[0] set to 0x00 (Accept All).
2. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period shall all be set to zero.
 3. The Lower Tester begins advertising using ADV_EXT_IND PDUs on the primary advertising channel and AUX_ADV_IND PDUs on the secondary advertising channel. The ADV_EXT_IND PDUs include an AuxPtr that refers to the AUX_ADV_IND PDUs on the secondary advertising channel. The AUX_ADV_IND PDUs include the AdvA field containing the Lower Tester address and a SyncInfo field referring to the AUX_SYNC_IND PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
 4. The Lower Tester generates AUX_SYNC_IND PDUs on the secondary advertising channel. The AUX_SYNC_IND PDUs contain the CTEInfo field, with CTETime set to any valid value, RFU set to '0', and CTEType set to a valid value for a CTE type that the IUT supports. Each packet containing an AUX_SYNC_IND PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
 5. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing an Advertising_SID and a nonzero Periodic_Advertising_Interval.
 6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Sync_Timeout shall be set to $3 \times$ Periodic_Advertising_Interval from Step 5. CTE_Type shall be set to a value that specifies to synchronize to packets with a CTE type from Step 4. The Upper Tester expects an HCI_Command_Status event in response.
 7. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 8. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to a valid value for a CTE type that the IUT supports.
 9. The Lower Tester ceases extended advertising but continues periodic advertising.
 10. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report from the IUT as in Step 8.
 11. The Lower Tester generates AUX_SYNC_IND PDUs as in Step 4, but with CTE_Type set to a valid value for a CTE type other than the value from Step 4.
 12. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to the same value from Step 11.
 13. The Lower Tester generates AUX_SYNC_IND PDUs as in Step 4, but without a CTEInfo field and a Constant Tone Extension.
 14. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT. CTE_Type shall be set to 0xFF (No Constant Tone Extension).
 15. Repeat Step 4. The Upper Tester expects to continue to receive HCI_LE_Periodic_Advertising_Report from the IUT as in Step 8.
 16. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to disable scanning.



- Expected Outcome

Pass Verdict

- The IUT maintains synchronization with the Lower Tester's periodic advertisement PDUs even after the Lower Tester ceases extended advertising.
- The IUT maintains synchronization with the Lower Tester's periodic advertisement PDUs after the Lower Tester or changes the Constant Tone Extension Type for the AUX_SYNC_IND PDUs or ceases transmitting a Constant Tone Extension with the AUX_SYNC_IND packet.

4.2.3.32 LL/DDI/SCN/BV-37-C [First AUX_SYNC_IND Never Received]

- Test Purpose

Tests that synchronization will fail on the IUT when an AUX_SYNC_IND PDU is not received within 6 periodic advertising events after the first advertising event is sent.

- Reference

[10] 4.4.3.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

The maximum number of octets the IUT can receive during scanning is defined in the "Scan Max Data" IXIT parameter.

- Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.



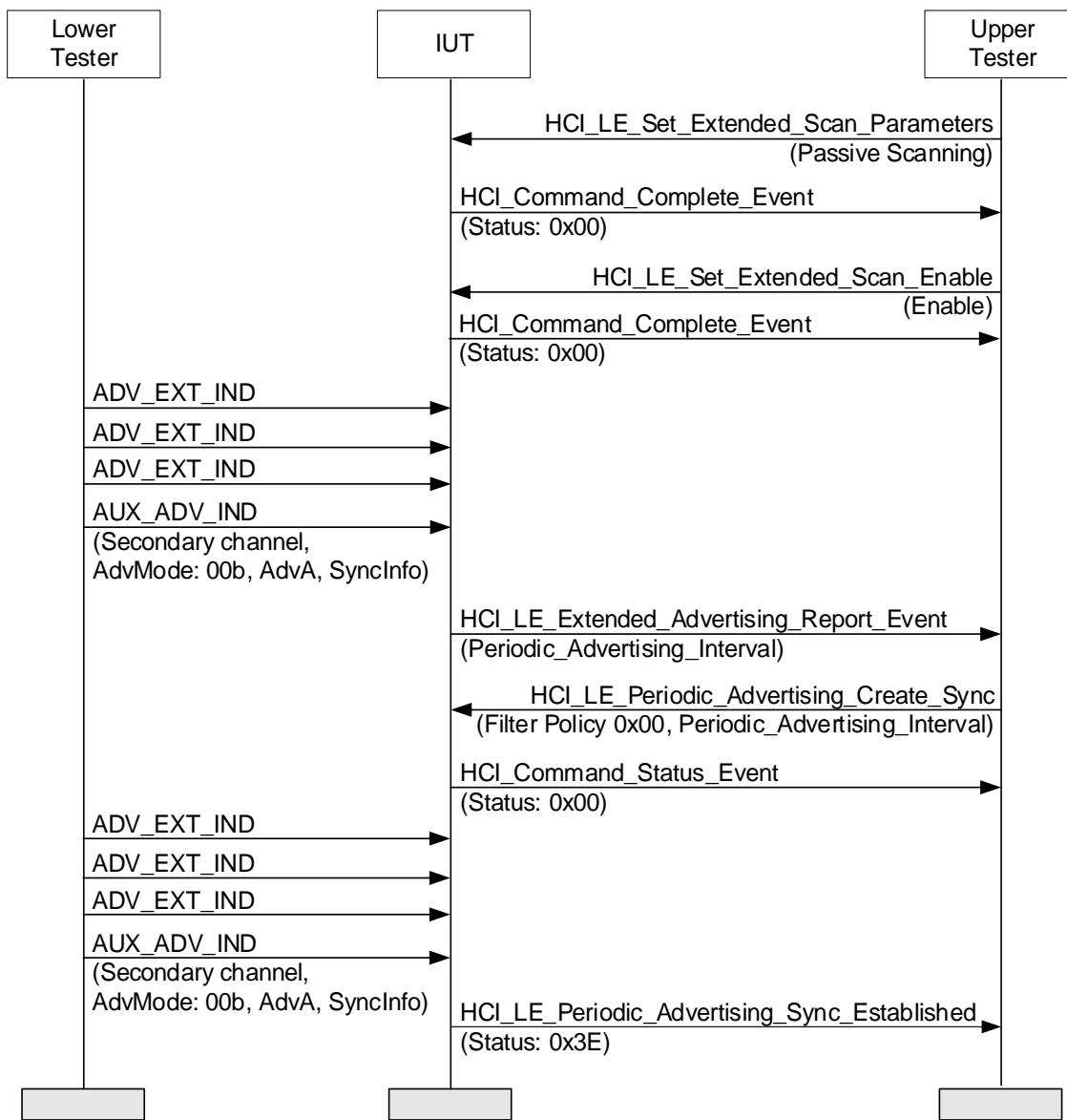


Figure 4.104: LL/DDI/SCN/BV-37-C [First AUX_SYNC_IND Never Received]

1. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Parameters` command to the IUT. The `Scanning_PHYs` parameter shall be set to 0x01 (LE 1M), `Scan_Type[0]` set to 0x00 (Passive Scanning), `Scan_Interval[0]` set to 0x0010, and `Scan_Window[0]` set to 0x0010. `Own_Address_Type` shall be set to 0x00 (Public Device Address), and `Scanning_Filter_Policy` shall be set to 0x00 (Accept All).
2. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Enable` command to the IUT to enable scanning. `Filter_Duplicates`, `Duration`, and `Period` shall all be set to zero.
3. The Lower Tester begins advertising using `ADV_EXT_IND` and `AUX_ADV_IND` PDUs using the LE 1M PHY. The `ADV_EXT_IND` PDUs shall include an `AuxPtr` that refers to the `AUX_ADV_IND` PDUs on the secondary advertising channel. The `AUX_ADV_IND` PDUs shall include the `AdvA` field containing the Lower Tester address and a `SyncInfo` field referring to a valid sequence of `AUX_SYNC_IND` PDUs. The Lower Tester does not transmit the `AUX_SYNC_IND` PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.

4. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval.
5. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT. Filter_Policy shall be set to 0x00 (Don't Use List). Advertising_SID shall be set to the Advertising_SID from Step 5. Advertiser_Address_Type shall be set to 0x00 (Public Device Address). Advertiser_Address shall be set to the Lower Tester's address. Skip shall be set to a valid Skip value. Sync_Timeout shall be set to (Skip + 3) x Periodic_Advertising_Interval from Step 4. Unused shall be set to 0x00. The Upper Tester expects an HCI_Command_Status event in response.
6. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event with Status set to 0x3E (Connection Failed To Be Established / Synchronization Timeout).

- Expected Outcome

Pass Verdict

The IUT ceases attempting to synchronize between 6 and 8 periodic advertising intervals after sending the HCI_Command_Status in step 6 and reports to the Upper Tester.

4.2.3.33 LL/DDI/SCN/BV-38-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled]

- Test Purpose

Tests that a scanner IUT can locate and receive periodic advertising events when reporting is initially disabled. The Lower Tester advertises using non-connectable and non-scannable extended advertising events with an AuxPtr field referring to a PDU on the secondary channel indicating the existence of periodic advertising, and the Upper Tester expects the IUT to report sync establishment without generating periodic advertising reports until reporting is enabled and stops generating reports after reporting is disabled.

- Reference

[13] 4.4.2.13.1, 4.4.3.4

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window) AND filtering policy to allow all devices (accept all advertising packets (0x00)).

- Test Procedure

Execute the test procedure with the IUT synchronizing to periodic advertisements generated by the Lower Tester.



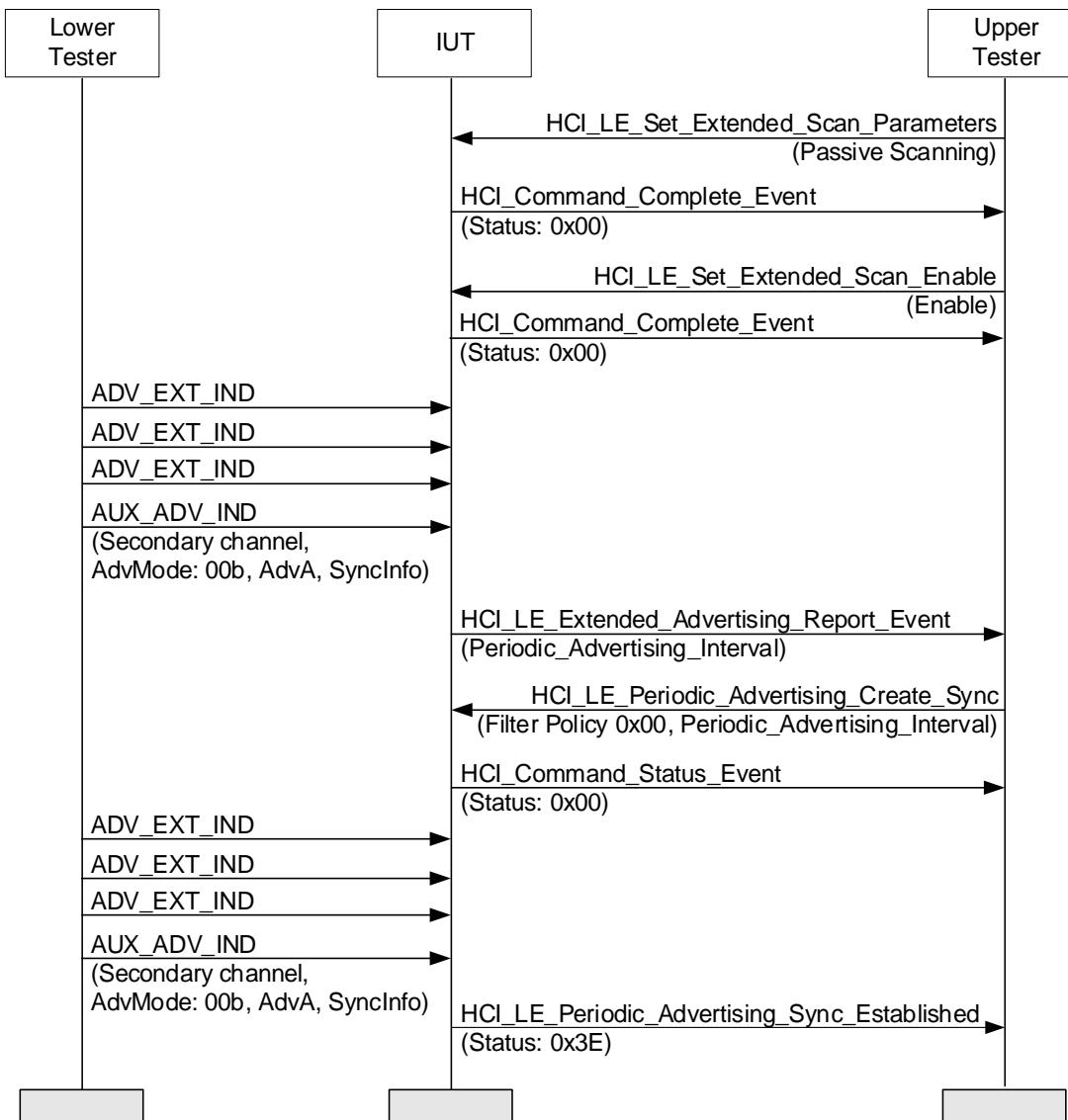


Figure 4.105: LL/DDI/SCN/BV-38-C [Extended Scanning, Periodic Advertising Reception, Reporting Initially Disabled]

1. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Parameters` command to the IUT with Scanning_PHYs set to 0x01 (LE 1M), Scan_Type[0] set to 0x00 (Passive Scanning), Scan_Interval[0] set to 0x0010, and Scan_Window[0] set to 0x0010. Own_Address_Type is set to 0x00 (Public Device Address), and Scanning_Filter_Policy is set to 0x00 (Accept All).
2. The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Enable` command to the IUT to enable scanning. Filter_Duplicates, Duration, and Period are all set to zero.
3. The Lower Tester begins advertising using `ADV_EXT_IND` and `AUX_ADV_IND` PDUs using the LE 1M PHY. The `ADV_EXT_IND` PDUs includes an AuxPtr that refers to the `AUX_ADV_IND` PDUs on the secondary advertising channel. The `AUX_ADV_IND` PDUs includes the AdvA field containing the Lower Tester address and a SyncInfo field referring to the `AUX_SYNC_IND` PDUs. The Lower Tester continues advertising until directed to stop in the test procedure.
4. The Lower Tester generates `ADV_SYNC_IND` PDUs on the secondary advertising channel using the indices selected by the LE Channel Selection Algorithm #2 as specified in the SyncInfo in

- Step 3. The Lower Tester continues periodic advertising until directed to stop in the test procedure.
5. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing a non-zero Periodic_Advertising_Interval.
 6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements. Options is set to 0x02 (Don't Use List, Reporting Initially Disabled). Advertising_SID is set to the Advertising_SID from Step 5. Advertiser_Address_Type is set to 0x00 (Public Device Address). Advertiser_Address is set to the Lower Tester's address. Skip is set to 0. Sync_Timeout is set to 3 x Periodic_Advertising_Interval from Step 5. CTE_Type is set to 0x00 (don't care). The Upper Tester expects an HCI_Command_Status event in response.
 7. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 8. The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
 9. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x01 (Reporting enabled) and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 10. The Upper Tester expects one or more HCI_LE_Periodic_Advertising_Report events from the IUT.
 11. Repeat Step 10 until at least 30 HCI_LE_Periodic_Advertising_Report events have been received.
 12. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x00 (Reporting disabled) and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 13. The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
 14. Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
 15. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 16. The Lower Tester ceases periodic advertising.
 17. If the IUT supports the LE Coded PHY, repeat steps 1-16, except that in Step 1 the Scanning_PHYs is set to 0x04 (LE Coded) and in steps 3-4 the Lower Tester advertises using the LE Coded PHY for both primary and secondary advertising channels.
 18. If the IUT supports the LE 2M PHY, repeat steps 1-16, except that in Step 1 the Scanning_PHYs is set to 0x01 (LE 1M) and in steps 3-4 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel.
- Expected Outcome
- Pass Verdict
- For all rounds described in the test procedure, the following conditions occur:
- The IUT generates HCI_LE_Periodic_Advertising_Sync_Established event containing a Status of 0x00 (Success) and synchronizes with the Lower Tester advertisements using the Channel Selection Algorithm #2.
 - The IUT does not generate HCI_LE_Periodic_Advertising_Report events when reporting is disabled.



- The IUT generates HCI_LE_Periodic_Advertising_Report events when reporting is enabled.
- The IUT terminates periodic advertising reception when the Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command.

4.2.3.34 LL/DDI/SCN/BI-01-C [Active Scanning Invalid CRC]

- Test Purpose

Tests that a scanner IUT ignores advertising indication packets with invalid CRC's.

The Lower Tester sends advertising packets with invalid CRC's and checks that the IUT does not send a scan request. The Lower Tester sends also scan response packets with invalid CRC's and observes the HCI events from the IUT.

- Reference

[3] 3.1, 4.4.3.2

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (public address, selected scan interval, selected scan window) AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner))

- Test Procedure

Execute the test procedure using the minimum advertising interval and the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval.



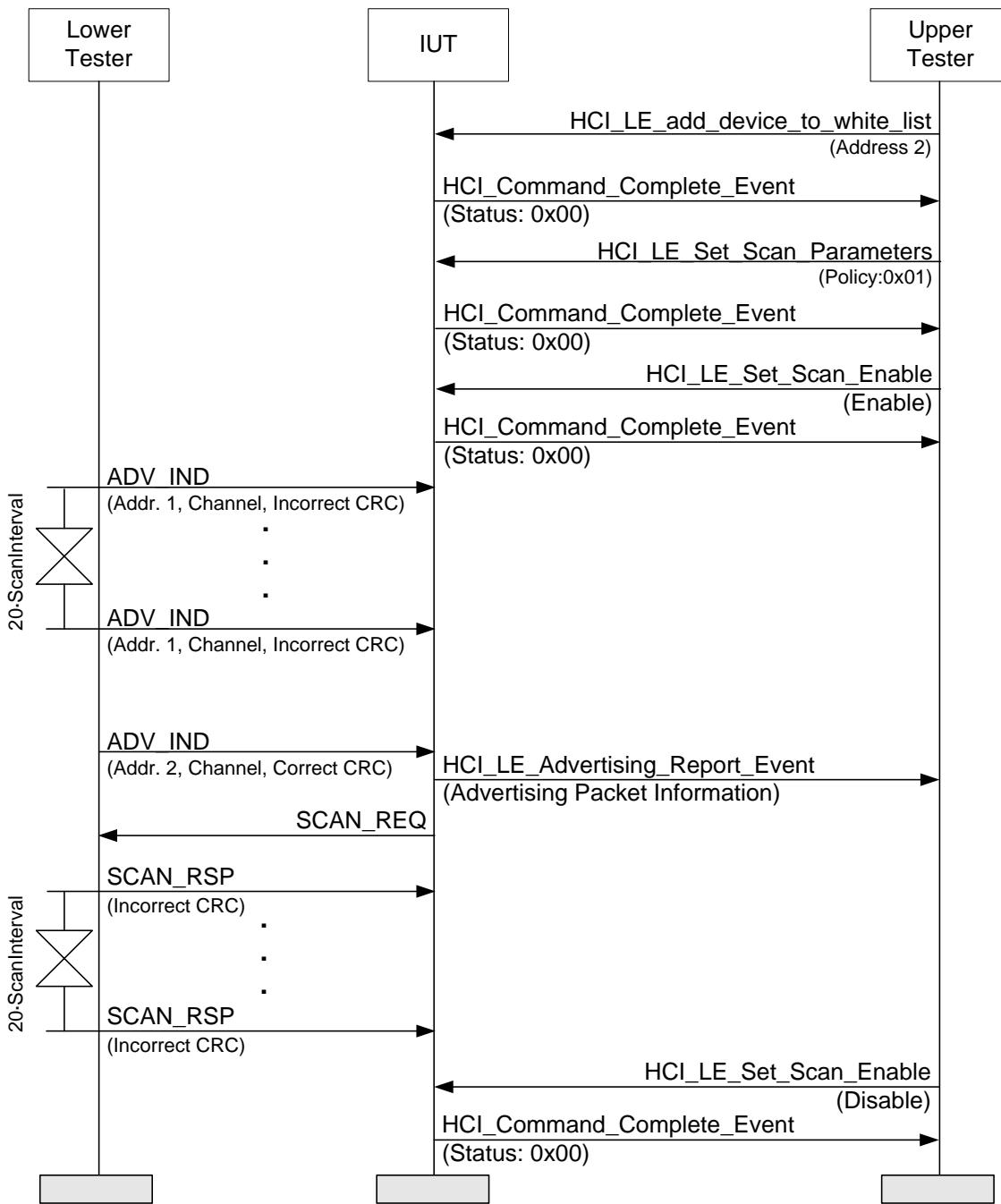


Figure 4.106: LL/DDI/SCN/BI-01-C [Active Scanning: Invalid CRC]

1. Upper Tester enables active scanning in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel.
3. Configure Lower Tester as advertiser using a common device address with invalid checksum.
4. Lower Tester sends ADV_IND packets with an invalid checksum, each advertising event, using the selected advertising interval. Repeat until the time exceeds a number of scan intervals (20).
5. Lower Tester expects no SCAN_REQ packets T_IFS after any of the ADV_IND packets.
6. Upper Tester expects no HCI_LE_Advertising_Report Event reporting the advertising packets sent by the Lower Tester in Step 4.
7. Configure Lower Tester as advertiser using a common device address (different than device address used in Step2) with correct checksum.

8. Lower Tester sends ADV_IND packets with correct checksum, each advertising event, using the selected advertising interval. Repeat until the time exceeds a number of scan intervals (20) or Step 9 executes.
9. Lower Tester expects an SCAN_REQ packet TIFS after any of the ADV_IND packets.
10. Lower Tester sends an SCAN_RSP packet with an invalid checksum to the IUT TIFS after the SCAN_REQ packet.
11. Interleave with Step 8, Upper Tester expects an HCI_LE_Advertising_Report Event containing the advertising packets information.
12. Interleave with Step 10. Upper Tester expects no HCI_LE_Advertising_Report Event for the response sent in Step 10.
13. Upper Tester sends an HCI_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
14. Configure Lower Tester advertising channel map using the second supported advertising channel.
15. Repeat steps 3-13.
16. Configure Lower Tester advertising channel map using the third supported advertising channel.
17. Repeat steps 3-13.

- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel with the timing combination, with the IUT ignoring the invalid advertising channel packets.

- Notes

In order to avoid interference, the test procedure uses the filtering policy to black list all unknown devices (0x01) if device filtering is supported. The features are assumed to be independent therefore the test can be executed without the filtering in conditions with LE traffic.

4.2.3.35 LL/DDI/SCN/BI-02-C [Passive Scanning Invalid CRC]

- Test Purpose

Tests that a scanner IUT ignores data in advertising packets with invalid checksums.

The Lower Tester sends advertising packets on several channels, expecting the IUT not to report the data with invalid checksums.

- Reference

[3] 3.1, 4.4.3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX,
LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window AND Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices) OR All White Listed (policy for scanner)).

- Test Procedure

Execute the test procedure with a minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).



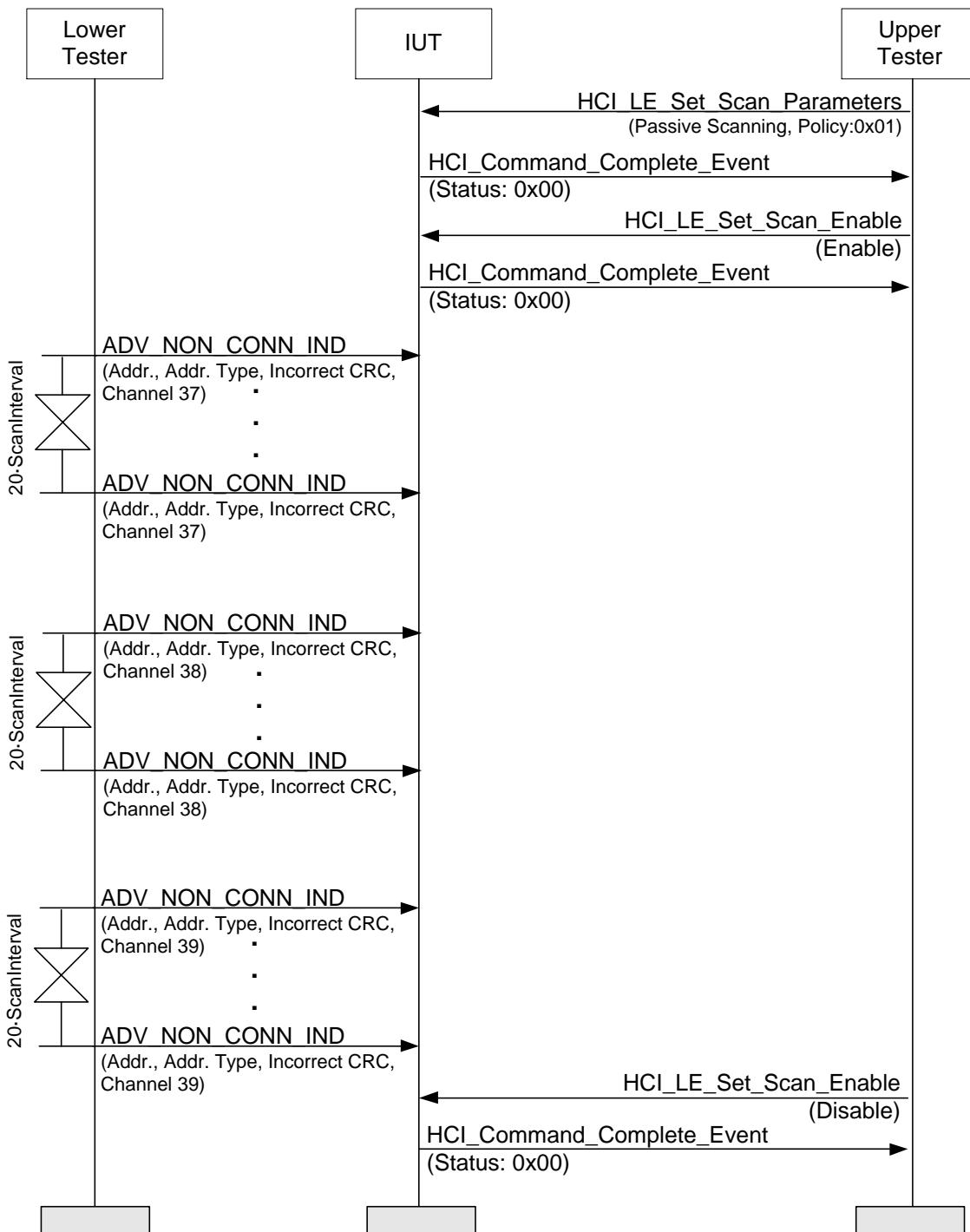


Figure 4.107: LL/DDI/SCN/BI-02-C [Passive Scanning: Invalid CRC]

1. Upper Tester enables passive scanning in the IUT.
2. Configure Lower Tester advertising channel map using the first supported advertising channel.
3. Configure Lower Tester as advertiser using a common device address with invalid checksum.
4. Lower Tester sends ADV_NONCONN_IND packets with an invalid checksum, each advertising event. Repeat for a time that exceeds a number of scan intervals (20).
5. Upper Tester expects no HCI_LE_Advertising_Report events containing the advertising packets information used in Step 4.



6. Configure Lower Tester advertising channel map using the second supported advertising channel.
7. Configure Lower Tester as advertiser using a common device address with invalid checksum.
8. Repeat steps 4-6.
9. Configure Lower Tester advertising channel map using the third supported advertising channel.
10. Configure Lower Tester as advertiser using a common device address with invalid checksum.
11. Repeat steps 4-6.
12. Issue an HCI_LE_Write_Scan_Enable to the IUT to stop the scanning function and expect an HCI_Command_Complete event in response.

- Expected Outcome

Pass Verdict

The test procedure completes using each supported advertising channel separately.

- Notes

In order to avoid interference, the test procedure uses the filtering policy to black list all unknown devices (0x01) if device filtering is supported. The features are assumed to be independent and the test can be executed without the filtering in conditions with LE traffic.

4.2.3.36 LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Wrong AdvA in Response]

- Test Purpose

Verify that the IUT when doing active scanning ignores a SCAN_RSP that does not contain the same AdvA used in the SCAN_REQ.

- Reference

[3] 4.4.3.2

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
 LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
 LL_initiator_Channel_Map

State: Active Scanning (resolvable private address, selected scan interval, selected scan window)
 AND (Specific White Listed (Lower Tester address, one public type address, policy for scanner, black list all unknown devices))



- Test Procedure

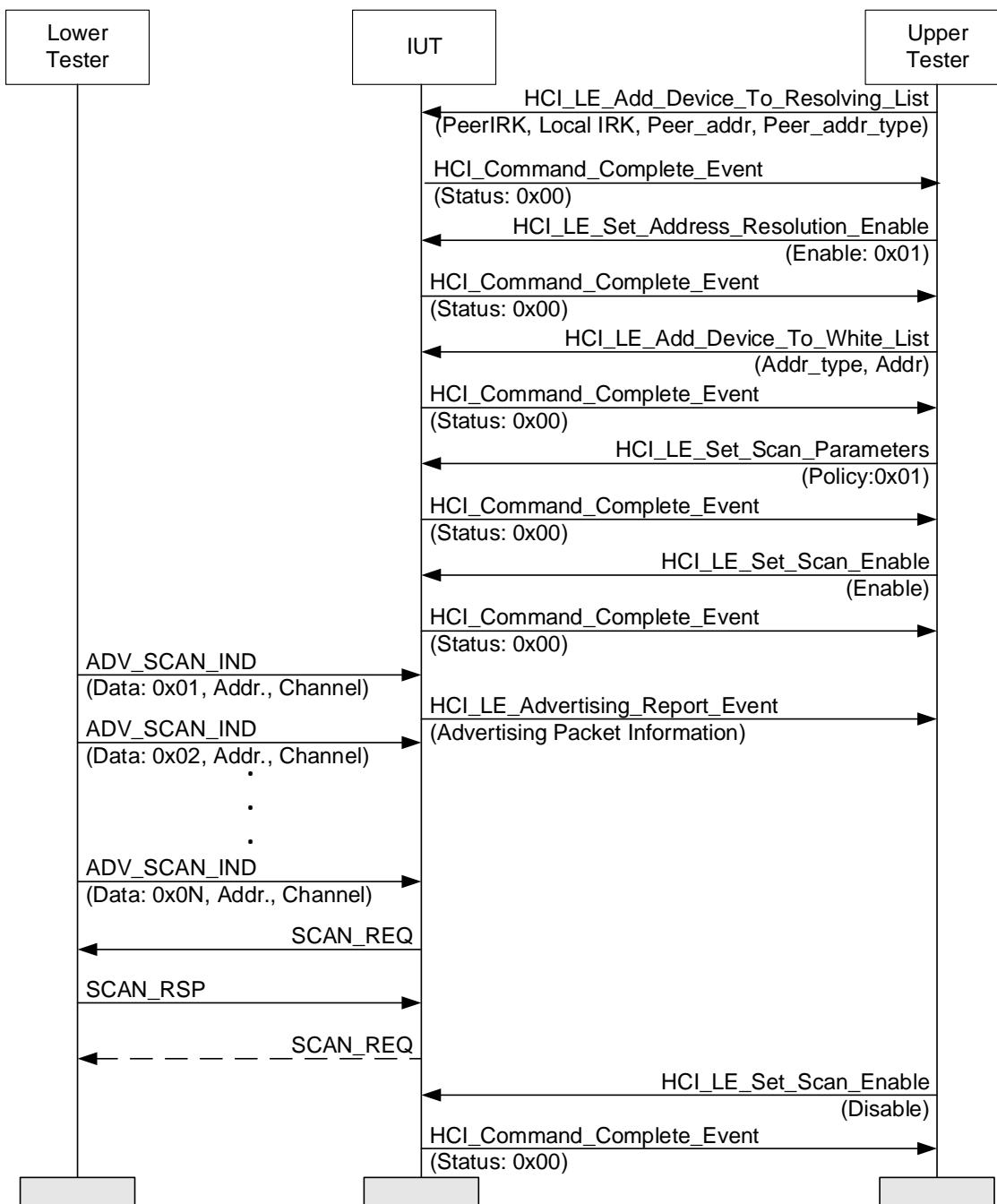


Figure 4.108: LL/DDI/SCN/BI-03-C [Privacy – Active Scanning, Local IRK, Peer IRK]

1. Configure the Lower Tester as an advertiser using a resolvable private address in the AdvA field.
 2. The Upper Tester adds peer device identity and local IRK information to resolving list. Using the HCI_LE_Add_Device_To_Resolving_List and HCI_LE_Set_Address_Resolution_Enable commands.
 3. The Upper Tester adds the address of the Lower Tester to the white list.

4. The Upper Tester enables active scanning with filtering policy set to 'Accept only advertising packets from devices where the advertiser's address is in the White List (0x01)' in the IUT.
5. The Lower Tester sends an ADV_SCAN_IND packet each advertising event using the selected advertising channel only. Repeat for at least 20 advertising intervals or until step 7 occurs.
6. The Lower Tester expects a SCAN_REQ packet TIFS after any of the ADV_SCAN_IND packets. The ScanA field in the SCAN_REQ packet shall use a resolvable private address.
7. The Lower Tester sends a SCAN_RSP packet TIFS after the SCAN_REQ packet. The AdvA field in the SCAN_RSP packet shall use a resolvable private address that does not match the one used in the SCAN_REQ packet. The AdvA field in the SCAN_RSP is generated using an IRK different from the IUT's IRK. The Upper Tester does not expect the IUT to send an HCI_LE_Advertising_Report event containing the information in the SCAN_RSP packet.
8. The Lower Tester optionally expects an additional SCAN_REQ packet from the IUT.
9. Interleave with Step 5: Upper Tester expects an HCI_LE_Advertising_Report containing the information used in the ADV_SCAN_IND packets.
10. The Upper Tester disables scanning.

- Expected Outcome

Pass Verdict

The IUT receives scannable undirected advertising using a resolvable private address in the AdvA field from the Lower Tester. The IUT successfully resolves the address.

The IUT responds to the advertising with a SCAN_REQ to the Lower Tester using a resolvable private address in the ScanA field. The Lower Tester successfully resolves the address.

The IUT receives a SCAN_RSP packet with an AdvA field that does not match the address from the SCAN_REQ packet. The IUT is expected not to send an HCI_LE_Advertising_Report event to the Upper Tester.

IUT stops scanning and sends HCI_Command_Complete event to the Upper Tester.

4.3 CON

Tests that the IUT behaves according to the connection setup and connection procedures.

4.3.1 Common PDU Contents

The packet descriptions for advertising and data channel packets sent and accepted by the Lower Tester are displayed below.

The addresses used in tests may vary for the Lower Tester,

The device address for the IUT is expected to match the IXIT value entered or set in the test preambles.

CONNECT_IND PDU:

Header					Length		Payload	
lsb msb					lsb msb		LSO MSO lsb msb	
Type '1010'	RFU '0'	ChSel '0'	TxAdd '0'	RxAdd '0'	Len '01000100'		Add (16 octets)	LLData (18 octets)



CONNECT_IND Payload Add Field:

LSO MSO lsb msb	LSO MSO lsb msb	LSO MSO lsb msb
InitA (6 octets)	AdvA (6 octets)	AA (4 octets)

CONNECT_IND Payload LLData Field:

The connection supervision timer value of 300 ms is based on the close to zero probability that an empty data packet would be received incorrectly around 10 (or more) times, with a slave device listening for every 3rd master transmission, given the BER of 0.1%.

CONNECTION UPDATE IND CtrData:

| LSO MSO
lsb msb |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| WinSize
(1 octet) | WinOffset
(2 octets) | Interval
(2 octets) | Latency
(2 octets) | Timeout
(2 octets) | Instant
(2 octets) |

FEATURE REQ CtrData:

LSO MSO
lsb msb

FeatureSet
(8 octets)

FEATURE RSP CtrData:

LSO MSO
lsb msb

FeatureSet
(8 octets)

4.3.2 ADV

Tests that the IUT behaves according to the connection setup procedures as an advertiser.

4.3.2.1 Common PDU Contents

CONNECT_IND Payload LL Data Field:

LSO MSO lsb msb	lsb msb	lsb msb	lsb msb	LSO MSO lsb msb	
CRCInit '10001001 00010001 10011000'	Interval '00010000'	Hop '10100'	SCA '000'	ChM '10000000 00000000 00000000 00000000 00000000'	RFU '000'

4.3.2.2 Common Variables

Master and slave roles in the connection handling tests use variables indicating the state of acknowledgements:

Name	Initial Value	Constraints	Description
current SN	Assigned from the SN in a packet sent.	The range is modulo 2.	The sequence number matching a packet sent.
current NESN	Assigned from the NESN in a packet sent.	The range is modulo 2.	The next expected sequence number to match a packet to be received.
next SN	0	The range is modulo 2, value current SN + 1.	The next sequence number to be sent.
next NESN	0	The range is modulo 2, value current NESN + 1.	The next expected sequence number to be sent.

The current values reflect a packet sent to the IUT. The values assigned to a packet are dependent on the test and the response packets from the IUT.

4.3.2.3 LL/CON/ADV/BV-01-C [Accepting Connections]

- Test Purpose

Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role, observing the packets and timing from the IUT.

- Reference

[3] 4.5.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices))



- Test Procedure

Execute the test procedure using the common data channel selection parameters and connection parameters to setup the connection.

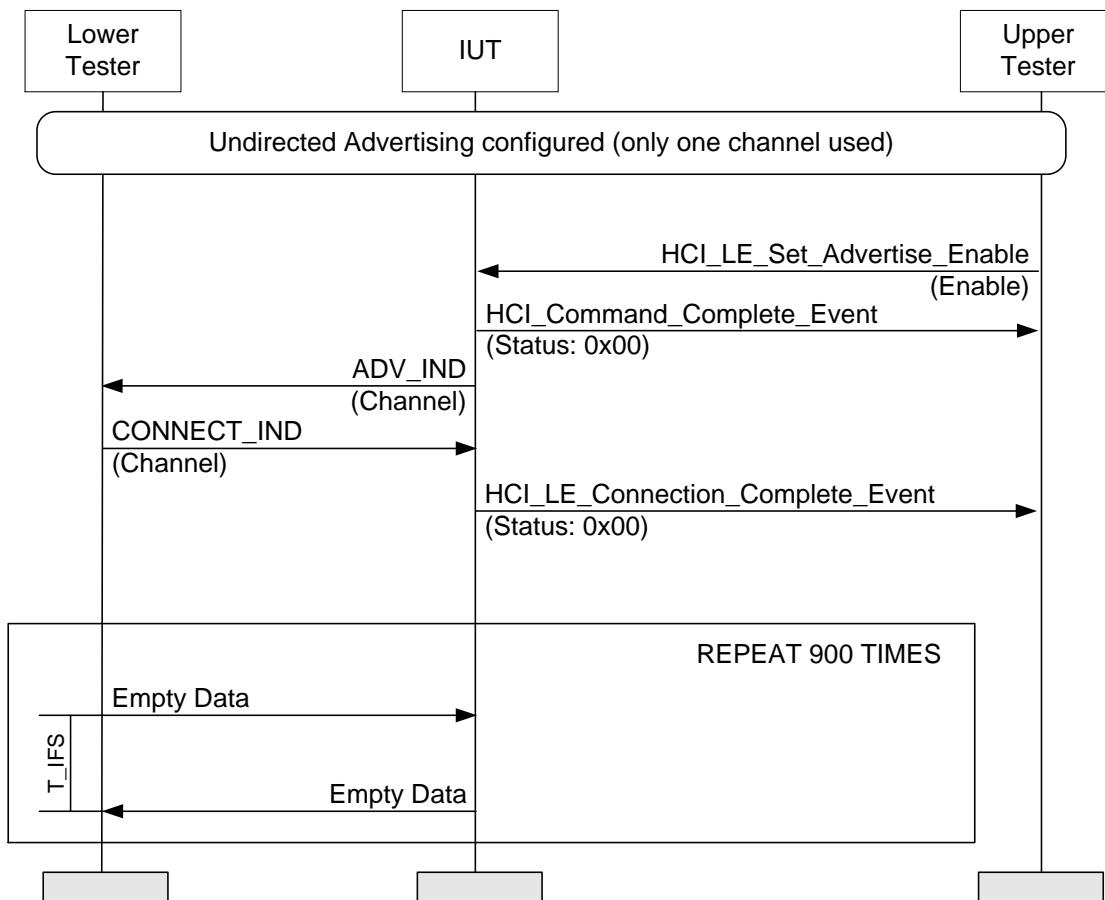


Figure 4.109: LL/CON/ADV/BV-01-C [Accepting Connections]

- Configure Lower Tester to initiate a connection.
- Upper Tester enables advertising in the IUT using the first supported advertising channel.
- Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only, using the selected advertising interval.
- Repeat 20 times or until the IUT stops advertising: Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
- Lower Tester sends a correctly formatted LL Data Channel PDU starting the first event at connection interval after the connection request using the common data channel selection parameters.
- Lower Tester expects a correctly formatted LL Data Channel PDU from the IUT T_IFS after the PDU sent on the same data channel. Lower Tester continues sending correctly formatted LL Data Channel PDUs until receiving a number of responses (100) to conclude the timing accuracy or up to 900 events transmitted.
- Interleave with Step 6: Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 4.
- Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).
- Configure Lower Tester to initiate a connection.



10. Upper Tester enables advertising in the IUT using the second supported advertising channel.
 11. Repeat steps 3-8.
 12. Configure Lower Tester to initiate a connection.
 13. Upper Tester enables advertising in the IUT using the third supported advertising channel.
 14. Repeat steps 3-8.
- Expected Outcome
- Pass Verdict
- The test procedure completes with the IUT stopping advertising by a connection requested from all advertising channels, moving to the data channel and responding in the connection events,
- The number of timing measurements for the reply packets from the IUT is at least 100 and the timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS,
- The IUT reports the connection setup with an HCI event.
- The IUT maintains the connection using the Channel Selection Algorithm #1.

- Notes
- The timing observation criterion takes into account active mode requirements. For the slave device the low power mode requirement is not tested here. The active mode timing in packet exchange is listed in the pass verdict to be within the 2 μ s range around T_IFS.
- The accuracy required is 1 μ s, from the expressions 150 μ s and 2 μ s. Drift does not affect measurements in exchanges of single data packets. Measurement accuracy is at least 0.1 μ s and rounding is done to microseconds.

In order to avoid interference, the test procedure uses the filtering policy to Allow Scan Request from White List, Allow Connect Request from White List if device filtering is supported. The features are assumed to be independent and the test can be executed without the filtering in conditions with low LE traffic.

4.3.2.4 LL/CON/ADV/BV-02-C [Accepting Connections Timeout]

- Test Purpose
- Tests that an advertiser IUT receives a connection request, stops advertising after the reception and after receiving no master transmission before connection supervision timer expiration, reports the connection setup failed.
- The Lower Tester acts in the initiating state, sending the connection request to the IUT then does not begin to maintain the connection and observes the IUT reports in HCI events.
- Reference
- [3] 4.5.5
- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices))



- Test Procedure

Execute the test procedure using the common data channel selection parameters and the common connection parameters to setup the connection.

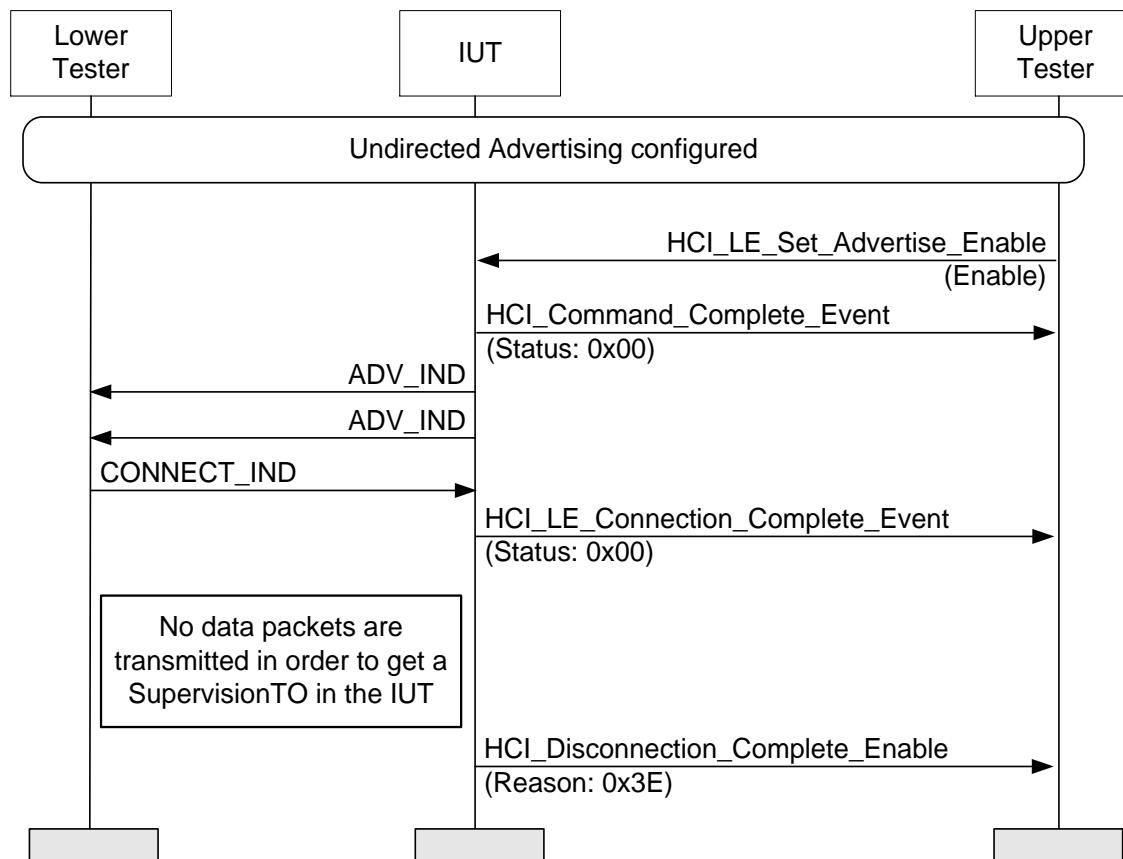


Figure 4.110: LL/CON/ADV/BV-02-C [Accepting Connections Timeout]

1. Configure Lower Tester to initiate a connection.
 2. Upper Tester enables advertising in the IUT using a selected supported advertising channel (defined as an IXIT).
 3. Repeat 20 times or until the IUT stops advertising: Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
 4. Lower Tester stops sending packets after CONNECT_IND packet and waits until connection supervision timeout timer expires.
 5. Interleave with Step 4: Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 3.
 6. Upper Tester expects an HCI_Disconnection_Complete event from the IUT with the reason parameter indicating 'connection failed to be established', with the connection handle parameter matching to Step 5.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT stopping advertising and reporting the failure to establish connection.



- Notes

The timeout accuracy is not observed here, but the parameter value may be used in test implementation.

4.3.2.5 LL/CON/ADV/BV-03-C [Master Missing Slave Packets]

- Test Purpose

Tests that an advertiser IUT after accepting a connection request, starts to maintain a connection in the slave role not taking slave latency into use before receiving an acknowledgement from the master.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role. The Lower Tester first sends a number of negative acknowledgements to the IUT, then changes to observe the acknowledgement scheme.

- Reference

[3] 4.5.5, 4.5.9

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

- Test Procedure

Execute the test procedure using the common data channel selection parameters and the common connection parameters to setup the connection, use a latency parameter value of 5.

If device filtering is supported, use the filtering policy to white list all unknown devices (apply the filtering policy of allowing scan request, connect request from any).



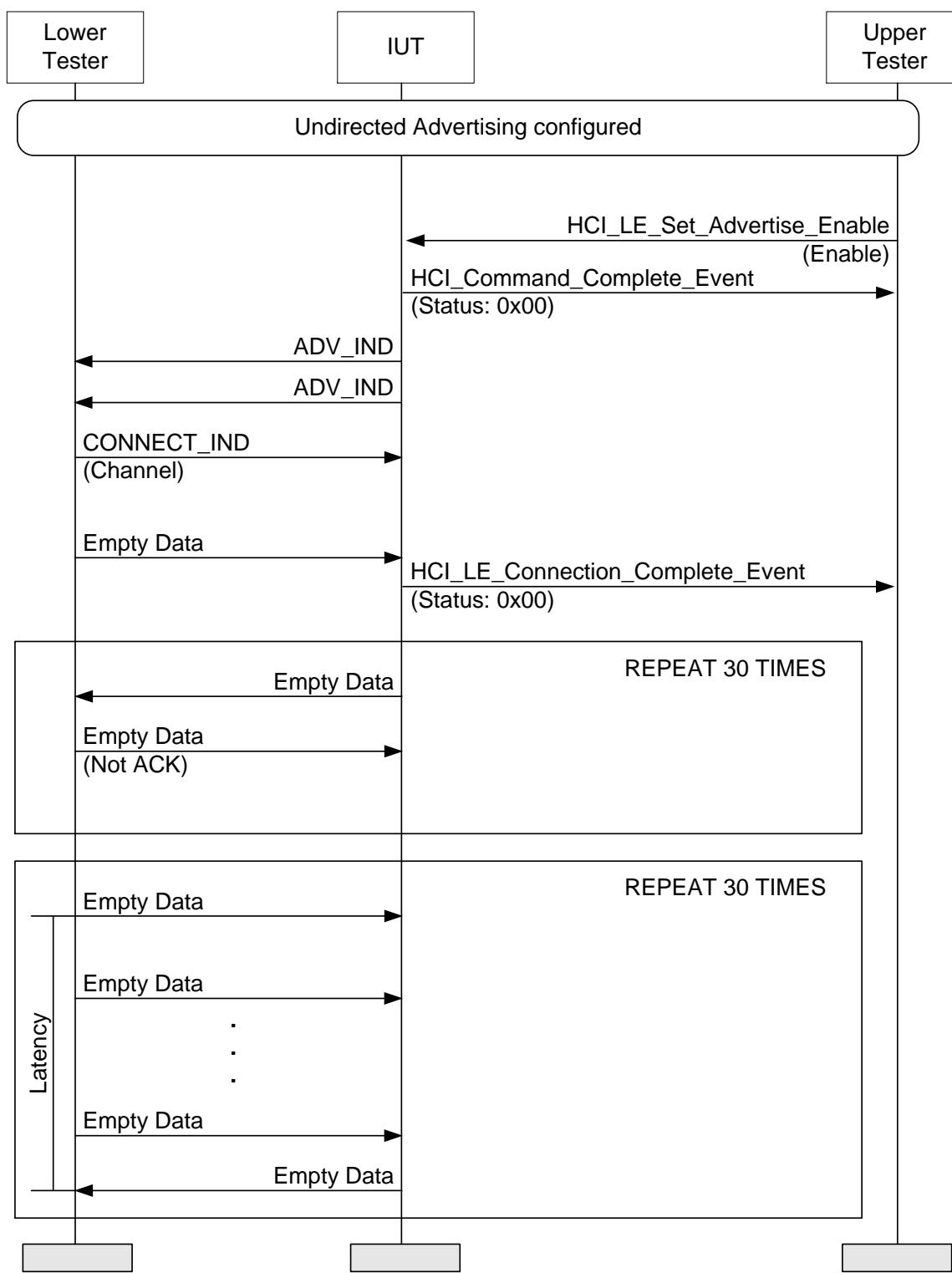


Figure 4.111: LL/CON/ADV/BV-03-C [Master Missing Slave Packets]

1. Configure Lower Tester to initiate a connection.
2. Upper Tester enables advertising in the IUT using a selected supported advertising channel (defined as an IXIT).
3. Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.



4. Lower Tester sends a correctly formatted LL Data Channel PDU, not acknowledging the slave packets, starting the first event using the common connection interval timing after the connection request.
5. Lower Tester expects a correctly formatted LL Data Channel PDU from the IUT, T_IFS after the PDU sent on the same data channel. Lower Tester continues sending correctly formatted LL Data Channel PDUs with the default timing.
6. Repeat steps 4-5 for a number of events (30).
7. Lower Tester sends a correctly formatted LL Data Channel PDU acknowledging the slave packets, using the common connection interval timing and the common data channel selection parameters.
8. Lower Tester expects a correctly formatted LL Data Channel PDU from the IUT, T_IFS after the PDU sent on the same data channel, for events between the common slave latency parameter
9. Repeat steps 7-8 for a number of events (30).
10. Interleave with steps 4-5: Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 3.
11. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from 10).

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT stopping advertising, moving to the data channel and responding in the connection events,

The IUT responds in at least 9 out of 30 connection events until the first master packet acknowledging the slave transmissions,

- Notes

The figure for the percentage of responses required is based on the correctly formatted LL Data Channel PDUs transmitted to and from the slave (around 85% probability of correct receptions).

4.3.2.6 LL/CON/ADV/BV-04-C [Directed Advertising Connection]

- Test Purpose

Tests that an advertiser IUT upon receiving a connection request to the directed advertising indications, stops advertising after the reception and starts to maintain a connection in the slave role.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role.

- Reference

[3] 4.4.2.4

- Initial Condition

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)



- Test Procedure

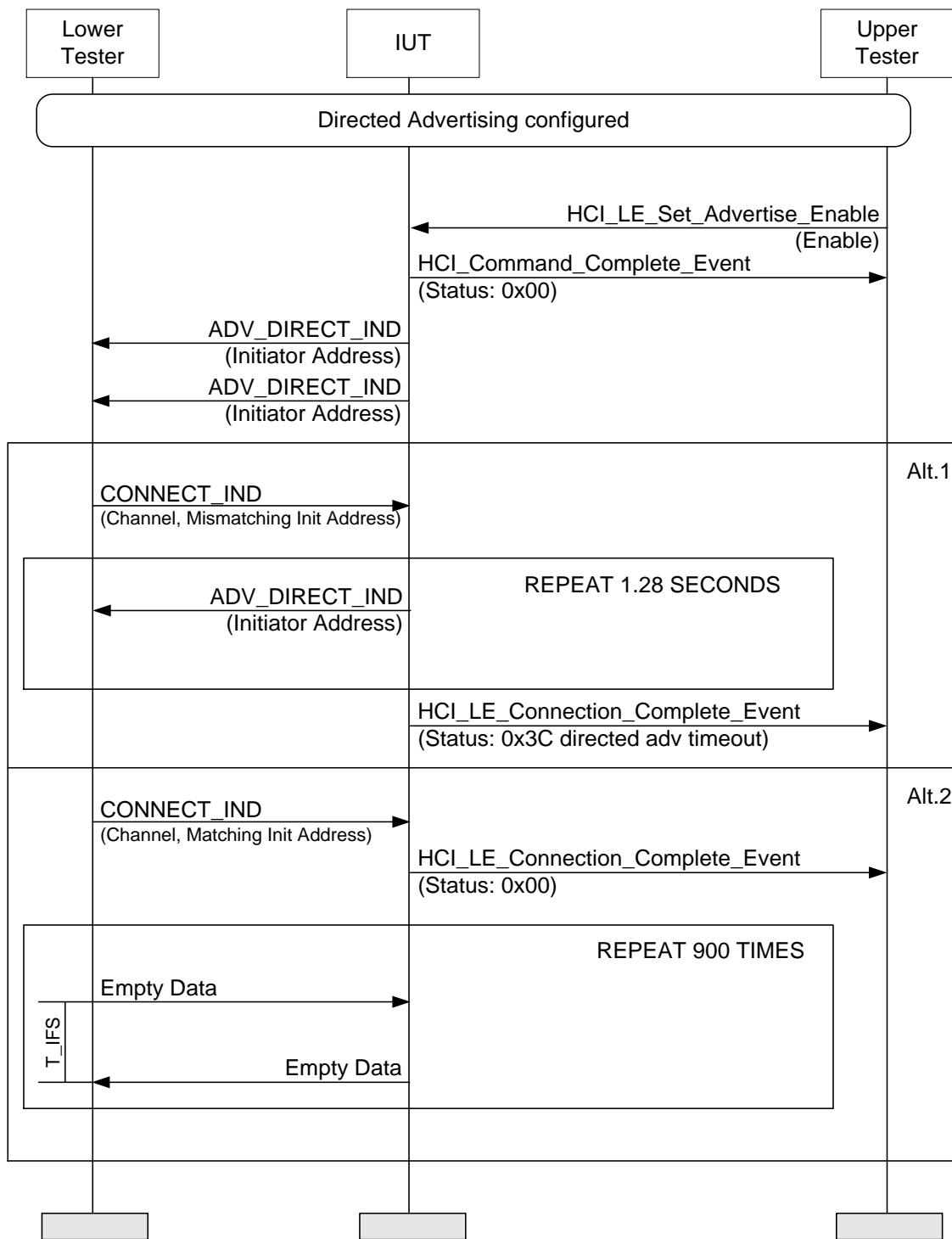


Figure 4.112: LL/CON/ADV/BV-04-C [Directed Advertising Connection]

- Configure Lower Tester to initiate a connection using a mismatching init address,
- Upper Tester enables directed advertising in the IUT using all supported advertising channels.
- Lower Tester expects an ADV_DIRECT_IND packet from the IUT and responds with a CONNECT_IND packet (with the mismatching init address) T_IFS after the end of the advertising packet.



4. Lower Tester expects ADV_DIRECT_IND packets after the connection request. Repeat up to a period of time (1.28 s) or until the IUT stops advertising.
 5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT with status parameter set to 'directed advertising timeout'.
 6. Configure Lower Tester to initiate a connection.
 7. Upper Tester enables directed advertising in the IUT using all supported advertising channels.
 8. Lower Tester expects a ADV_DIRECT_IND packet from the IUT and responds with a CONNECT_IND packet (with a matching init address) TIFS after the end of the advertising packet
 9. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 8.
 10. Lower Tester sends a correctly formatted LL Data Channel PDU starting the first event at connection interval after the connection request using the common data channel selection parameters.
 11. Lower Tester expects a correctly formatted LL Data Channel PDU from the IUT, TIFS after the correctly formatted LL Data Channel PDU sent on the same data channel and continues sending correctly formatted LL Data Channel PDUs until receiving a number of responses (900).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT first rejecting, then accepting the connection request and maintaining the connection in the slave role,

The IUT reports the connection created with an HCI event.

The IUT maintains the connection using the Channel Selection Algorithm #1.

4.3.2.7 LL/CON/ADV/BV-05-C [Extended Advertising, Accepting Connections]

- Test Purpose

Tests that an advertiser IUT, using connectable advertising, receives a connection request on the secondary channel, stops advertising after the reception, and starts to maintain a connection in the slave role. The Lower Tester acts first in the initiating state, sending the connection request to the IUT on the secondary channel, and then starts to maintain a connection in the master role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.

- Reference
- [10] 4.4.2.3, 4.3.2, 4.4.2.8, 4.4.2
- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

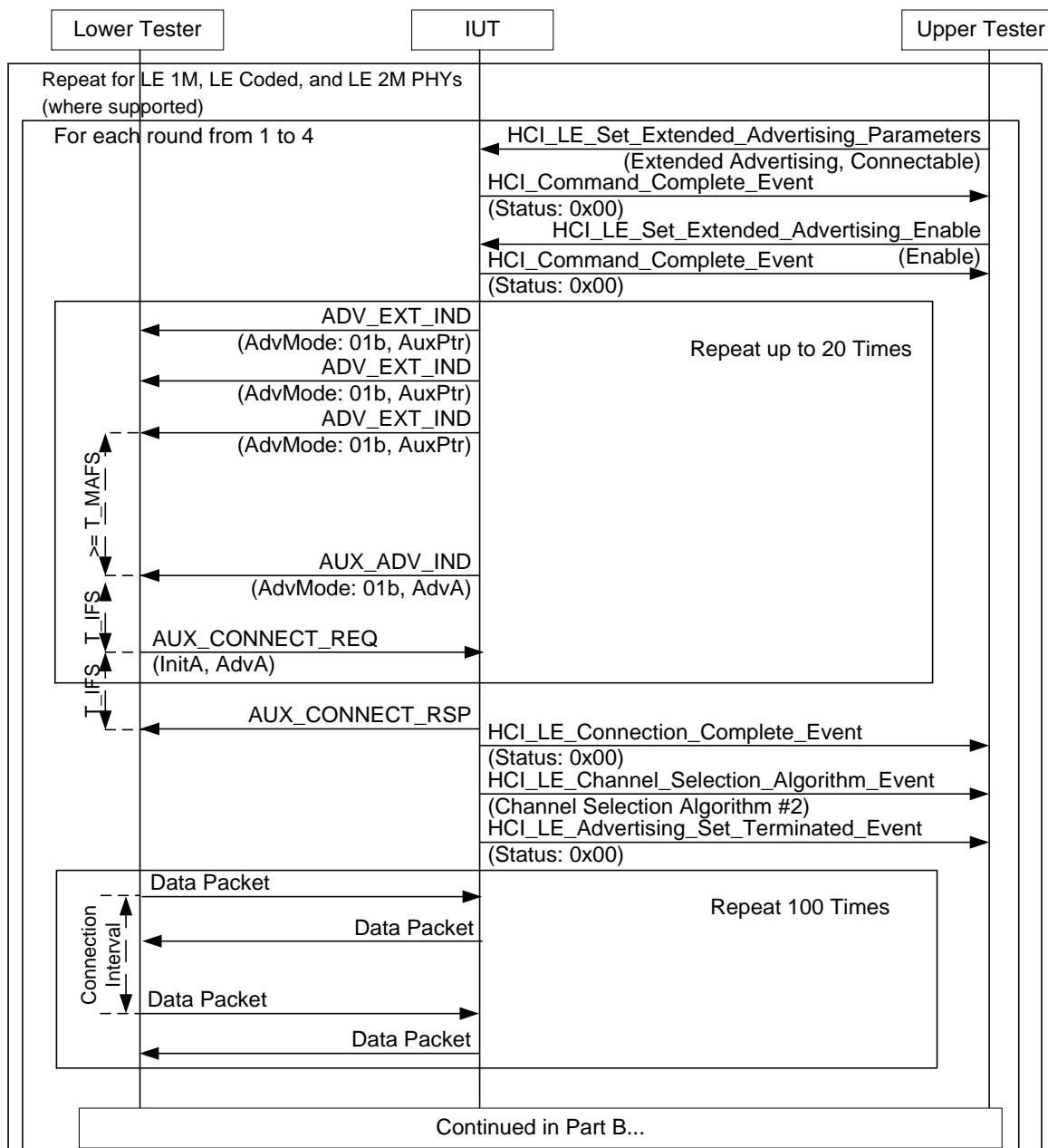


Figure 4.113: LL/CON/ADV/BV-05-C [Extended Advertising, Accepting Connections – Part A]



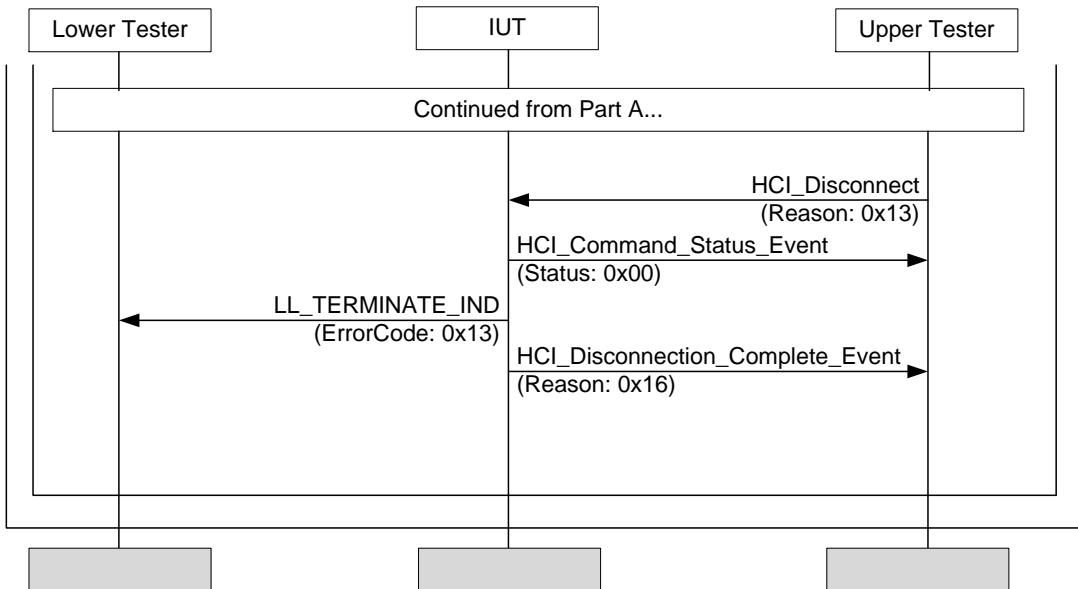


Figure 4.114: LL/CON/ADV/BV-05-C [Extended Advertising, Accepting Connections – Part B]

For each round from 1 to 4 Based on [Table 4.27](#):

1. The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command to the IUT using all supported advertising channels and minimum advertising interval. `Advertising_Event_Properties` parameter shall be set to the value specified in [Table 4.27](#) for this round. The `Own_Address_Type` shall be set to 0x00 (Public Device Address). The `Primary_Advertising_PHY` shall be set to 0x01 (LE 1M). The `Secondary_Advertising_PHY` shall be set to 0x01 (LE 1M). The `Advertising_Filter_Policy` shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the `HCI_LE_Set_Extended_Advertising_Enable` command. The `Duration[0]` parameter is set to 0x0000 (No Advertising Duration). The `Max_Extended_Advertising_Events[0]` parameter is set to 0x00 (No Max).
3. The Lower Tester expects an `ADV_EXT_IND` packet from the IUT with `AdvMode` set to 01b with only the `AuxPtr` Extended Header field.
4. The Lower Tester utilizes the `AuxPtr` field to listen for an `AUX_ADV_IND` PDU on the secondary advertising channel with the `AdvMode` field set to 01b and with the `Adva` Extended Header field present. The `AdvA` field shall contain the IUT's Advertising Address. If this round used directed advertising, `AUX_ADV_IND` PDU shall also include the `TargetA` Extended Header field set to the Lower Tester's Public Device Address. The Lower Tester responds with an `AUX_CONNECT_REQ` PDU on the secondary advertising channel `T_IFS` after receiving the `AUX_ADV_IND` PDU. The `InitA` field shall be set to the Lower Tester's address. The `AdvA` field shall be set as shown in [Table 4.27](#) for the current Round.
5. If the `AUX_CONNECT_REQ` PDU has an `AdvA` not equal to the IUT's address, the Lower Tester expects no `AUX_CONNECT_RSP` packet from the IUT. The Upper Tester disables advertising using the `HCI_LE_Set_Extended_Advertising_Enable` command. Skip to Step 15.
6. The Lower Tester expects an `AUX_CONNECT_RSP` PDU from the IUT on the secondary advertising channel `T_IFS` after sending the `AUX_CONNECT_REQ`. If no `AUX_CONNECT_RSP` PDU is received, repeat steps 3-6 up to 20 times. If the Lower Tester does not receive that PDU after 20 times, this test case ends with a Fail Verdict. After receiving an `AUX_CONNECT_RSP` PDU, verify that the IUT has started to maintain a connection by responding to the Lower Tester's LL data packets.
7. The Lower Tester expects no further `ADV_EXT_IND` PDUs after the advertising interval from the IUT. The Lower Tester continues scanning for advertising packets during the free time while



maintaining the connection in steps 10-13. If the Lower Tester receives a further advertising packet, the test ends with a Fail Verdict.

8. The Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, unencrypted, connection handle). Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
9. The Upper Tester expects an HCI_LE_Advertising_Set_Terminated event from the IUT with the Status set to 0x00, the Advertising_Handle from Step 1, and the Connection_Handle from Step 8.
10. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.
11. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
12. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
13. Repeat steps 10–12 for a number of events (100 events).
14. The Upper Tester terminates the connection.
15. Repeat steps 1–14 for each Round shown in [Table 4.27](#).
16. If the IUT supports the LE Coded PHY, repeat steps 1–15, except that in Step 1 the Primary_Advertising_PHY shall be set to 0x03 (LE Coded) and the Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
17. If the IUT supports the LE 2M PHY, repeat steps 1–15, except that in Step 1 the Primary_Advertising_PHY shall be set to 0x01 (LE 1M) and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).

Round	HCI_LE_Set_Extended_Advertising_Parameters (Step 1)	AUX_CONNECT_REQ PDU (Step 4)
	Advertising_Event_Properties	AdvA
1	0x0001	IUT
2	0x0001	Not IUT
3	0x0005	IUT
4	0x0005	Not IUT

Table 4.27: Payload contents for each case variation

- Expected Outcome

Pass Verdict

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

- The IUT sends an ADV_EXT_IND with an AuxPtr field referring to an AUX_ADV_IND on the secondary advertising channel.
- If the advertising was directed, the AUX_ADV_IND has a TargetA field containing the Lower Tester's address.
- The IUT responds to the AUX_CONNECT_REQ within the 2 μ s range around T_IFS.
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_MAFS.
- If the AUX_CONNECT_REQ PDU has an AdvA not equal to the IUT's address then the IUT shall not respond.



- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports termination of advertising with an HCI LE Advertising Set Terminated event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data on the expected PHY selected in Step 1.

4.3.2.8 LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections]

- Test Purpose

Tests that an advertiser IUT, using undirected connectable advertising with legacy PDUs, receives a connection indication on the primary channel, stops advertising after the reception, and starts to maintain a connection in the slave role. The Lower Tester acts first in the initiating state, sending the connection request to the IUT on the primary channel, and then starts to maintain a connection in the master role. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

The Lower Tester requests a connection from the IUT, and then checks that advertising has stopped.

- Reference

[10] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

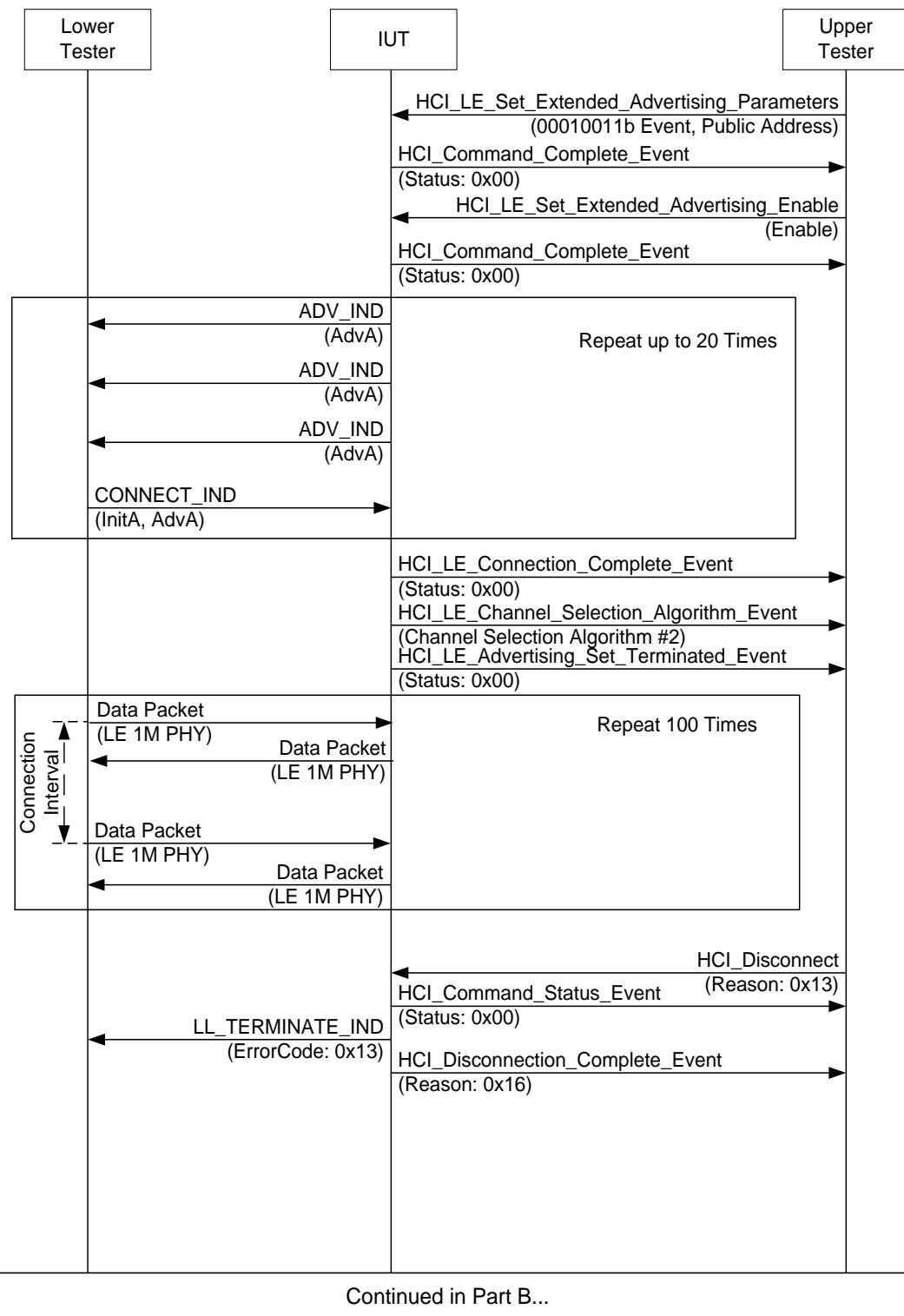


Figure 4.115: LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections – Part A]



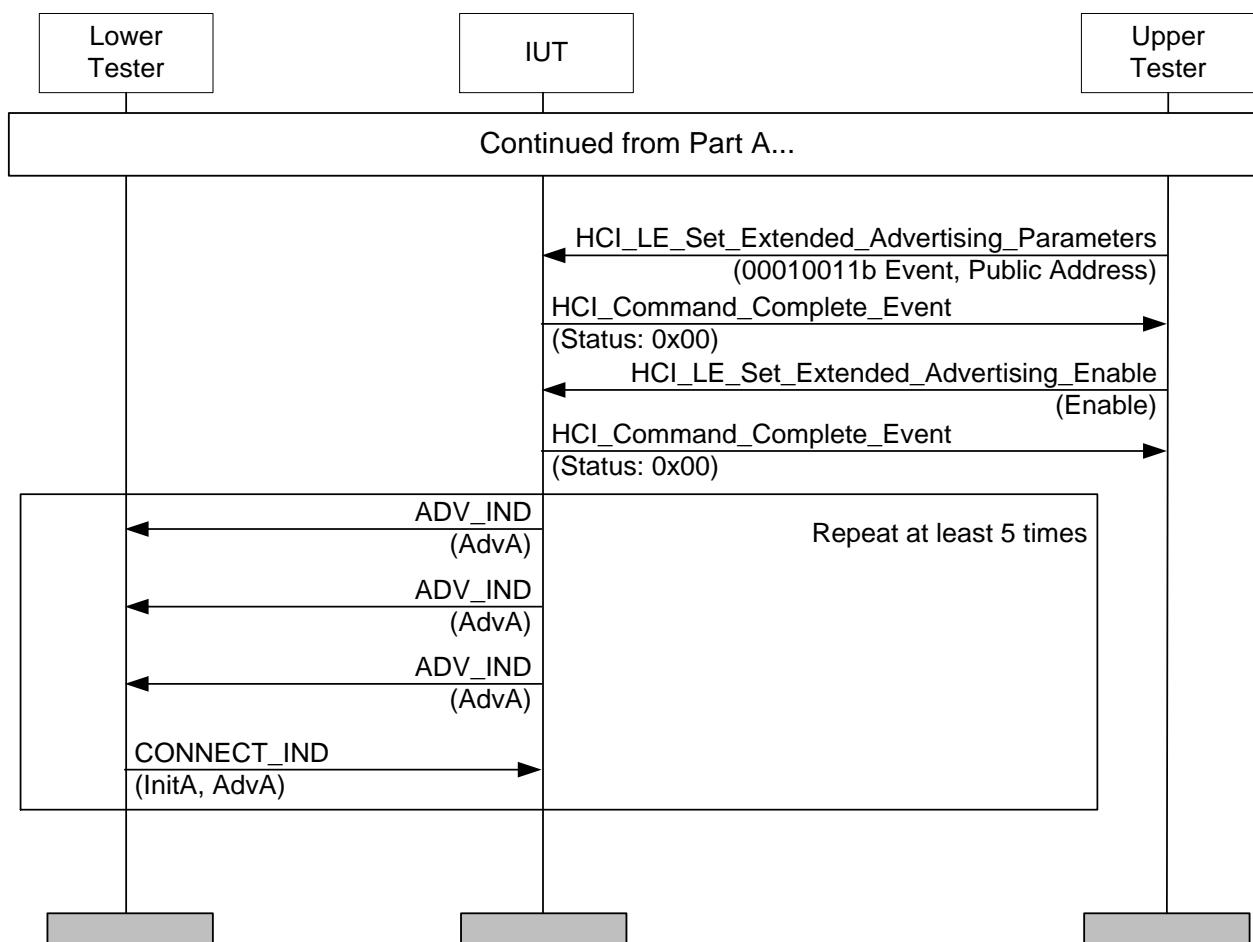


Figure 4.116: LL/CON/ADV/BV-06-C [Extended Advertising, Legacy PDUs, Accepting Connections – Part B]

1. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter shall be set to 00010011b (ADV_IND legacy PDU). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration). The Max_Extended_Advertising_Events[0] parameter is set to 0x00 (No Max).
3. The Lower Tester expects the IUT to send ADV_IND, with ChSel set to 1 (Channel Selection Algorithm #2), on the advertising channel. The AdvA field shall contain the IUT's Advertising Address.
4. The Lower Tester responds with a CONNECT_IND PDU, with ChSel set to 1 (Channel Selection Algorithm #2), on the primary advertising channel. The InitA field shall be set to the Lower Tester's address. The AdvA field shall be set to the IUT's address.
5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). If the Upper Tester does not receive the event, repeat steps 3-5 up to 20 times. If the Upper Tester does not receive the event after 20 times, this test case ends with a Fail Verdict.
6. The Lower Tester continues scanning for advertising packets during the free time while maintaining the connection in steps 9-10. If the Lower Tester receives a further advertising packet, the test ends with a Fail Verdict.



7. Immediately after receiving the HCI_LE_Connection_Complete event, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
 8. The Upper Tester expects an HCI_LE_Advertising_Set_Terminated event from the IUT with the Status set to 0x00, the Advertising_Handle from Step 1, and the Connection_Handle from Step 5.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat step 9 for a number of events (100 events).
 11. The Upper Tester terminates the connection.
 12. Repeat steps 1 to 4, except that in step 4 the AdvA field shall be set to an address other than the IUT's address.
 13. The Lower Tester expects the IUT to continue sending ADV_IND PDUs for at least 5 advertising intervals and responds with a CONNECT_IND PDU each time.
- Expected Outcome

Pass Verdict

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

- In step 4, the IUT sends an ADV_IND, with ChSel set to 1, on the primary advertising channel using the LE 1M PHY.
- In step 5, the IUT reports the requested connection with an HCI LE Connection Complete event.
- In step 7, the IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- In step 8, the IUT reports termination of advertising with an HCI LE Advertising Set Terminated event.
- In step 9, the IUT sends and receives data using the LE 1M PHY.
- In step 13, if the CONNECT_IND PDU has an AdvA not equal to the IUT's address then the IUT shall not respond.

4.3.2.9 LL/CON/ADV/BV-07-C [Accepting Connections, Channel Selection Algorithm #1]

- Test Purpose

Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates no support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to zero (0), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #1. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

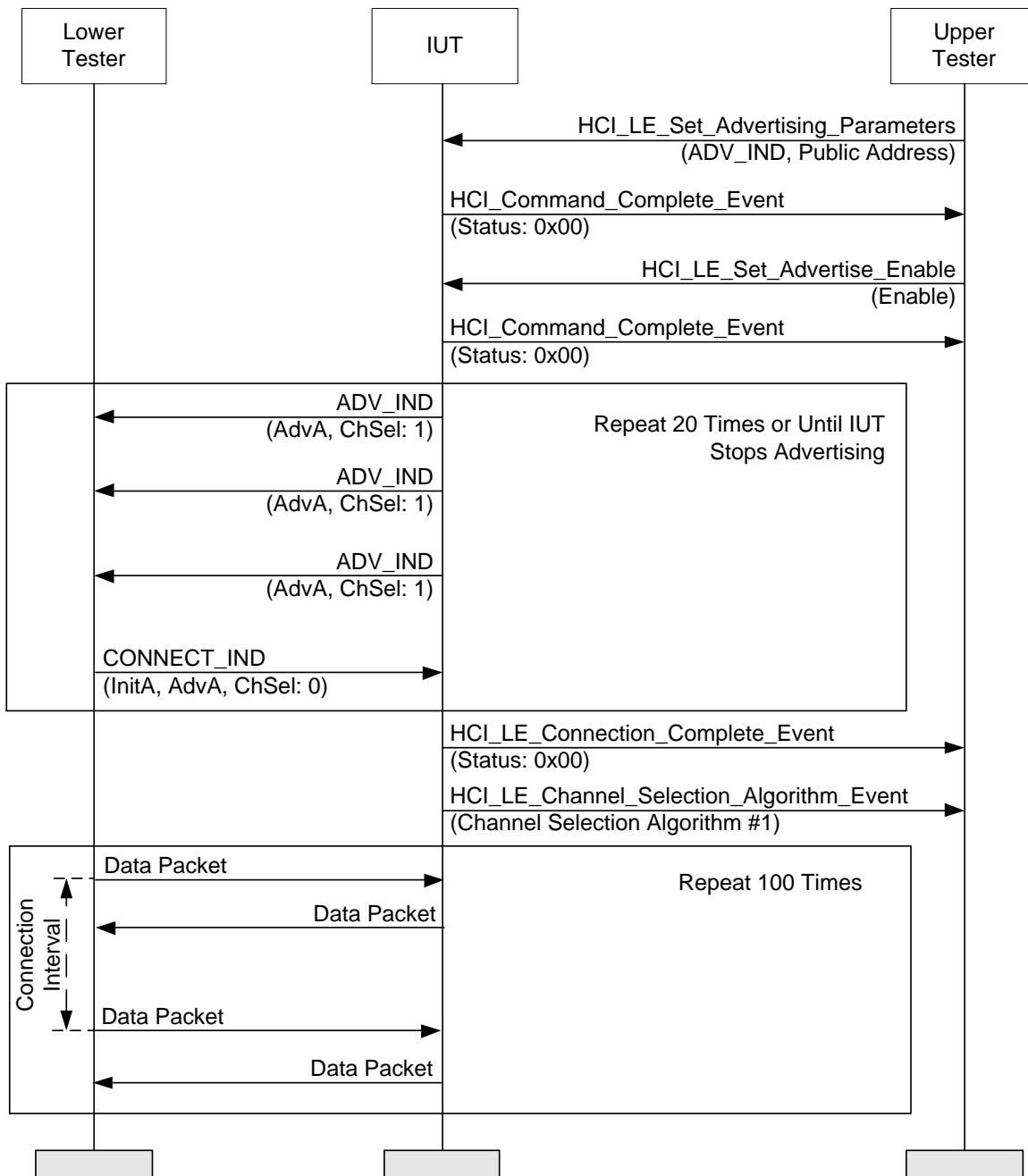


Figure 4.117: LL/CON/ADV/BV-07-C [Accepting Connections, Channel Selection Algorithm #1]

- The Upper Tester sends an **HCI_LE_Set_Advertising_Parameters** command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
- The Upper Tester enables advertising using the **HCI_LE_Set_Advertise_Enable** command.
- The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 1 on the advertising channel. The AdvA field shall contain the IUT's Advertising Address.
- The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 0. The InitA field shall be set to the Lower Tester's address, and the AdvA field shall be set to the IUT's Advertising Address.

5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4-5 up to 20 times or until the IUT stops advertising.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.
 8. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat steps 7-9 for a number of events (100 events).
- Expected Outcome

Pass Verdict

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

- The IUT sends ADV_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

4.3.2.10 LL/CON/ADV/BV-08-C [Directed Advertising Connection, Channel Selection Algorithm #1]

- Test Purpose

Tests that an advertiser IUT receives a connection request to the directing advertising indication, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates no support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to zero (0), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #1. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference
[\[10\]](#) 4.4.2.4, 4.3.2
- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.



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

- Test Procedure

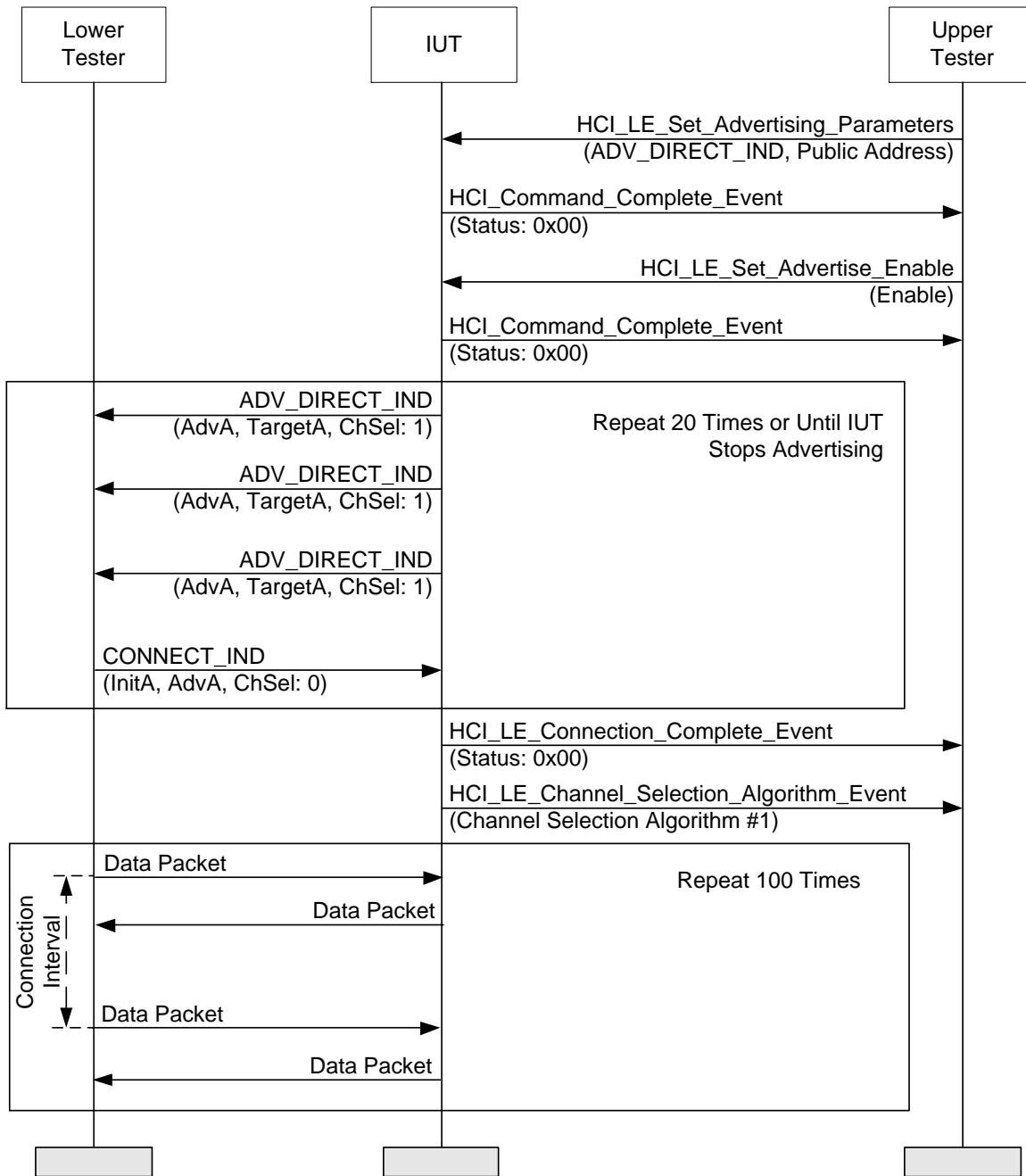


Figure 4.118: LL/CON/ADV/BV-08-C [Directed Advertising Connection, Channel Selection Algorithm #1]

1. The Upper Tester sends an `HCI_LE_Set_Advertising_Parameters` command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x01 (`ADV_DIRECT_IND`). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address shall be set to the Lower Tester's address. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the `HCI_LE_Set_Advertise_Enable` command.



3. The Lower Tester expects the IUT to send ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT's Advertising Address. The TargetA field shall contain the Lower Tester's address.
 4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 0. The InitA field shall be set to the Lower Tester's address, and the AdvA field shall be set to the IUT's Advertising Address.
 5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4-5 up to 20 times or until the IUT stops advertising.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.
 8. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat steps 7-9 for a number of events (100 events).
- Expected Outcome

Pass Verdict

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

- The IUT sends ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

4.3.2.11 LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2]

- Test Purpose

Tests that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #2. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.4.2.3, 4.3.2



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

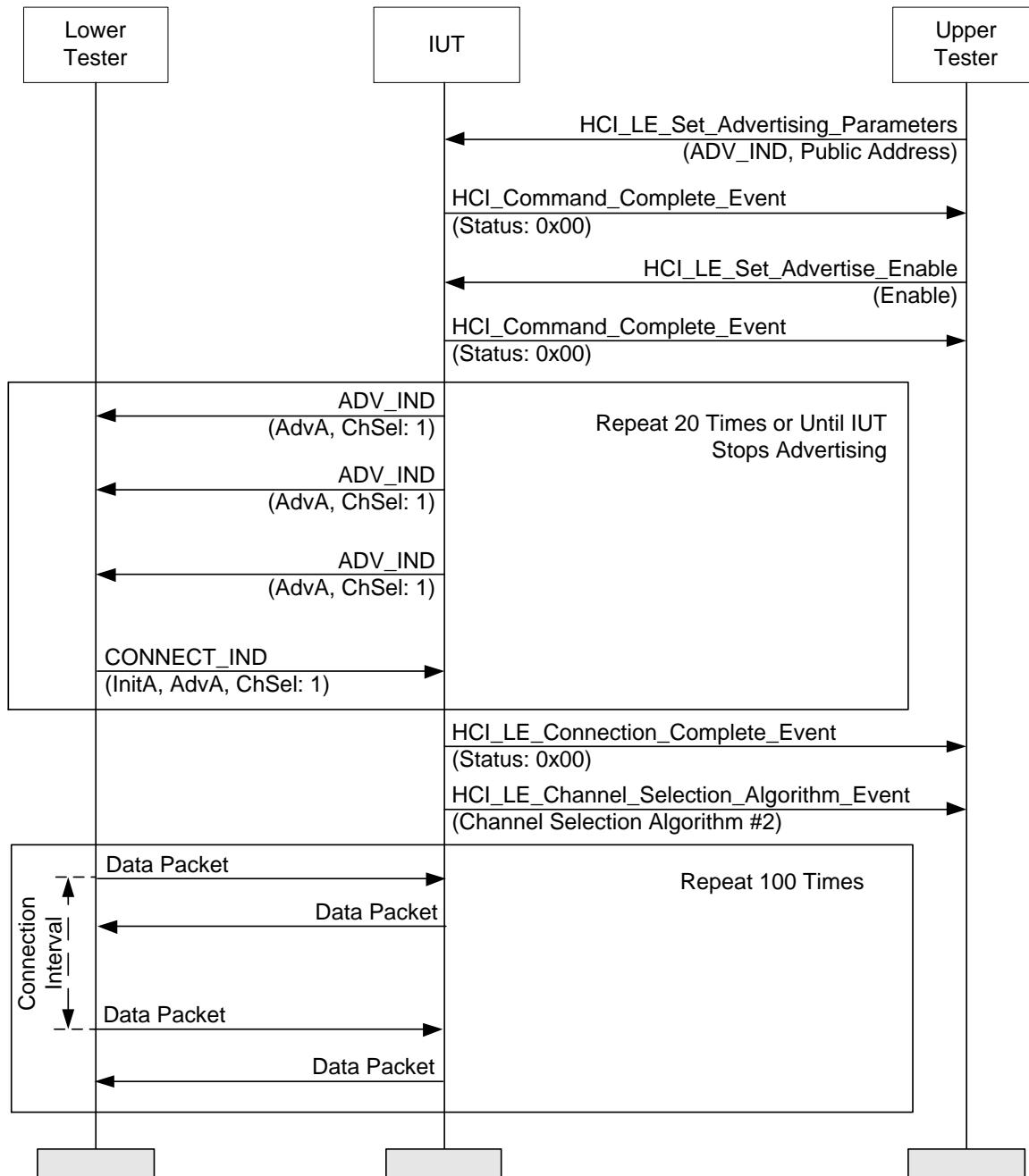


Figure 4.119: LL/CON/ADV/BV-09-C [Accepting Connections, Channel Selection Algorithm #2]

- The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set



- to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the HCI_LE_Set_Advertise_Enable command.
 3. The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT's Advertising Address.
 4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester's address, and the AdvA field shall be set to the IUT's Advertising Address.
 5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #2. If no data packets are received, repeat steps 4-5 up to 20 times or until the IUT stops advertising.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.
 8. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat steps 7-9 for a number of events (100 events).
- Expected Outcome

Pass Verdict

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

- The IUT sends ADV_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

4.3.2.12 LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2]

- Test Purpose

Tests that an advertiser IUT receives a connection request to the directing advertising indication, stops advertising after the reception and starts to maintain a connection in the slave role when the connection request indicates support for Channel Selection Algorithm #2. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role, observing the packets and timing from the IUT to confirm the IUT is using the Channel Selection Algorithm #2. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.



- Reference

[10] 4.4.2.4, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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

- Test Procedure

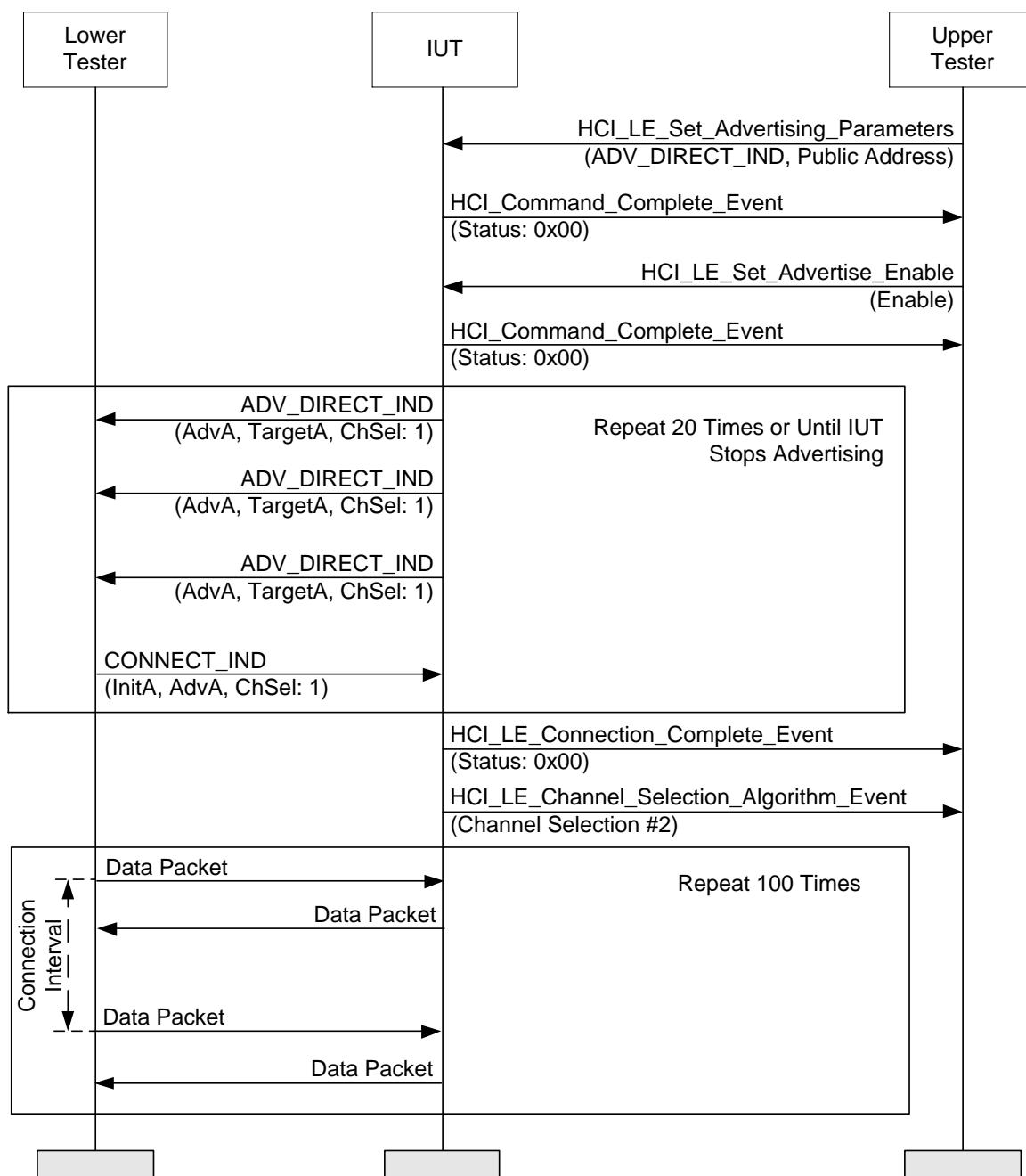


Figure 4.120: LL/CON/ADV/BV-10-C [Directed Advertising Connection, Channel Selection Algorithm #2]



1. The Upper Tester sends an HCI_LE_Set_Advertising_Parameters command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x01 (ADV_DIRECT_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address_Type shall be set to 0x00 (Public Device Address). The Peer_Address shall be set to the Lower Tester's address. The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
 2. The Upper Tester enables advertising using the HCI_LE_Set_Advertise_Enable command.
 3. The Lower Tester expects the IUT to send ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel. The AdvA field shall contain the IUT's Advertising Address. The TargetA field shall contain the Lower Tester's address.
 4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester's address, and the AdvA field shall be set to the IUT's Advertising Address.
 5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's corrected formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #2. If no data packets are received, repeat steps 4-5 up to 20 times or until the IUT stops advertising.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle). Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2.
 8. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat steps 7-9 for a number of events (100 events).
- Expected Outcome

Pass Verdict

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

- The IUT sends ADV_DIRECT_IND PDUs with ChSel set to 1 on the primary advertising channel.
- The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
- The IUT reports the requested connection with an HCI LE Connection Complete event.
- The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.
- The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

4.3.2.13 LL/CON/ADV/BI-01-C [Connection Supervision Timeout during Fail Connection Setup]

- Test Purpose

Verifies that an advertiser IUT correctly implements the connection supervision timeout, under a condition where it expires during connection setup.



The Lower Tester acts first in the initiating state, sending the connection request to the IUT, and then starts to maintain a connection in the master role but with packets with invalid checksums.

- Reference

[3] 4.5.5, 4.5.9

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND (Specific White Listed (Lower Tester address, one public type address, policy for advertiser, black list all unknown devices) OR All White Listed (policy for advertiser))

- Test Procedure

Execute the test procedure using the common data channel selection parameters and the common connection interval to maintain the connection.

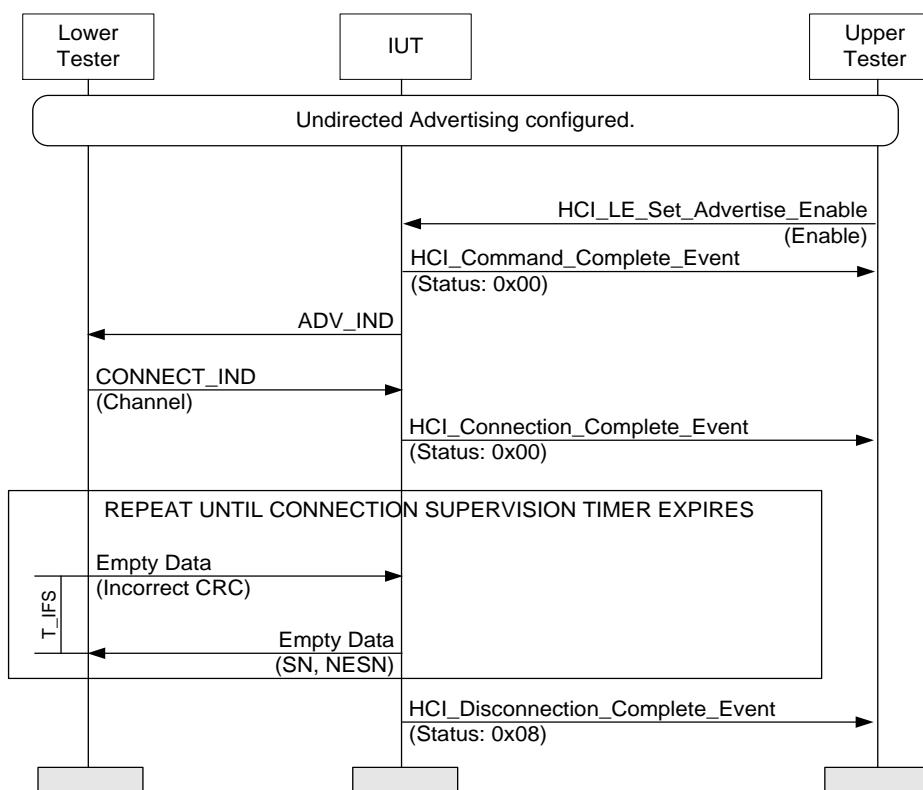


Figure 4.121: LL/CON/ADV/BI-01-C [Connection Supervision Timeout during fail connection setup]

1. Configure Lower Tester to initiate a connection and send packets with incorrect CRC after CONNECT_IND packet,
2. Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
3. Lower Tester expects the IUT to send ADV_IND packets, on the selected advertising channel only.
4. Lower Tester responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.



5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in Step 4
 6. Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum, starting the first event using the common connection interval timing after the connection request and using the common data channel selection parameters.
 7. Lower Tester expects a correctly formatted LL Data Channel PDU, not acknowledging the invalid packet, from the IUT, TIFS after the PDU sent on the same data channel.
 8. Repeat steps 6-7 until connection supervision timer expires
 9. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, with status parameter set to 'connection timeout' and with the connection handle matching to Step 5.
- Expected Outcome

Pass Verdict

The IUT responds to the master transmissions in at least 3 out of 6 events.

4.3.2.14 LL/CON/ADV/BV-11-C [Accepting Connections, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2]

- Test Purpose

Tests that an advertiser IUT that only supports Channel Selection Algorithm #1 receives a connection request from a Lower Tester that supports Channel Selection Algorithm #2. The IUT stops advertising after the reception and starts to maintain a connection in the slave role with Channel Selection Algorithm #1. The Lower Tester acts first in the initiating state, sending the connection request to the IUT with ChSel set to one (1), then starts to maintain a connection in the master role. The IUT observes the packets and timing from the Lower Tester to confirm the Lower Tester is using the Channel Selection Algorithm #1.

- Reference

[10] 4.4.2.3, 4.3.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

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



- Test Procedure

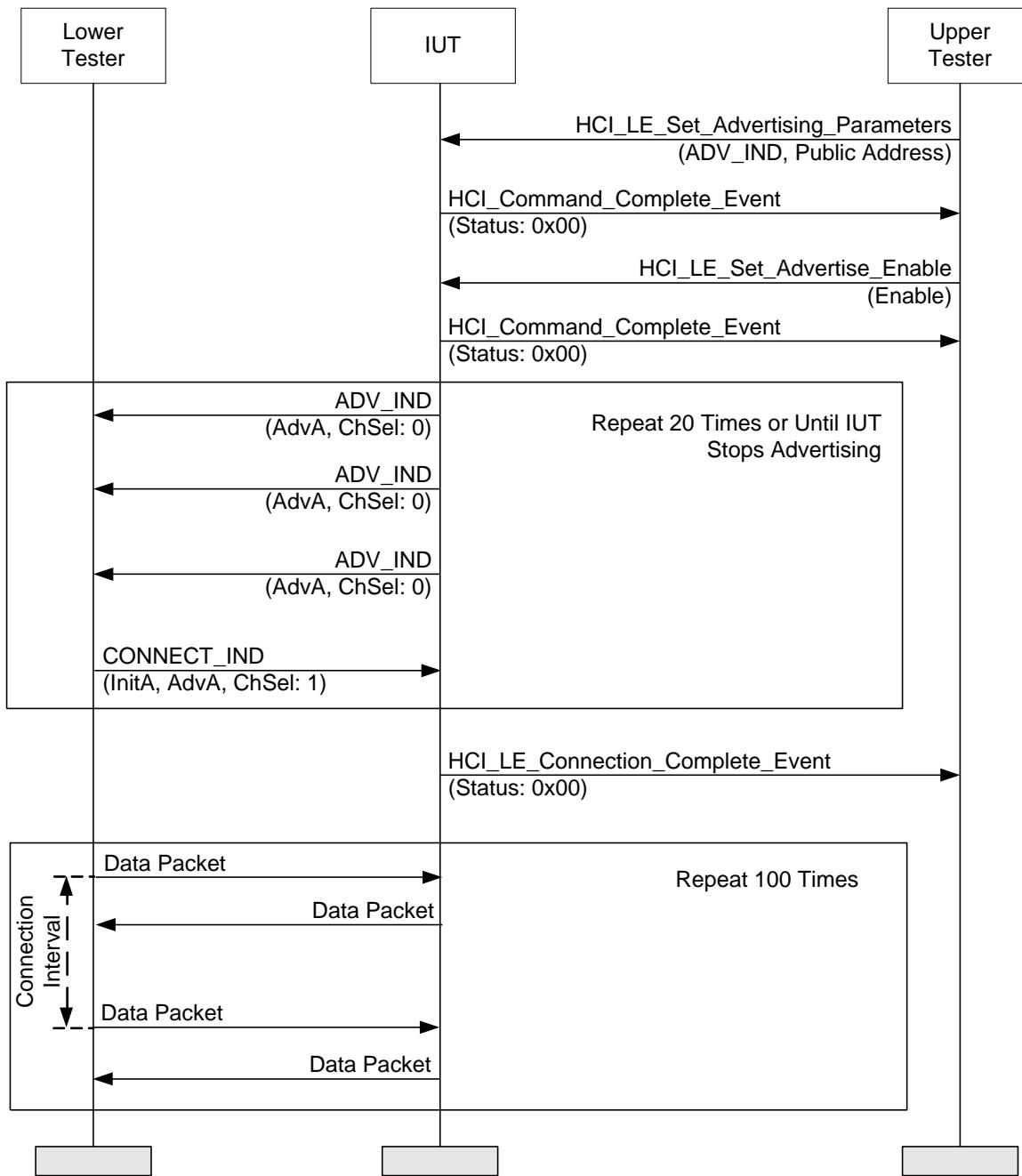


Figure 4.122: LL/CON/ADV/BV-11-C [Accepting Connections, Channel Selection Algorithm #1]

1. The Upper Tester sends an **HCI_LE_Set_Advertising_Parameters** command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Type shall be set to 0x00 (ADV_IND). The Own_Address_Type shall be set to 0x00 (Public Device Address). The Advertising_Filter_Policy shall be set to 0x00 (accept connection requests from all devices).
2. The Upper Tester enables advertising using the **HCI_LE_Set_Advertise_Enable** command.
3. The Lower Tester expects the IUT to send ADV_IND PDUs with ChSel set to 0 on the advertising channel. The AdvA field shall contain the IUT's Advertising Address.
4. The Lower Tester responds with a CONNECT_IND PDU with ChSel set to 1. The InitA field shall be set to the Lower Tester's address, and the AdvA field shall be set to the IUT's Advertising Address.



5. The Lower Tester verifies that the IUT has started to maintain a connection by responding with correctly formatted LL Data Channel PDUs to the Lower Tester's correctly formatted LL Data Packets on the data channels selected by LE Channel Selection Algorithm #1. If no data packets are received, repeat steps 4-5 up to 20 times or until the IUT stops advertising.
 6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT and as postamble: Slave Connection (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1.
 8. The Lower Tester expects a correctly formatted LL Data Channel PDU using the acknowledgement scheme, from the IUT on the same data channel.
 9. The Lower Tester sends correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals.
 10. Repeat steps 7-9 for a number of events (100 events).
- Expected Outcome
- Pass Verdict
- For all rounds described in the test procedure, the following condition shall occur:
- The IUT sends ADV_IND PDUs with ChSel set to 0 on the primary advertising channel.
 - The IUT receives CONNECT_IND PDUs with ChSel set to 1.
 - The test procedure completes with the IUT stopping advertising using an advertising interval between the minimum and maximum on all of the supported advertising channels.
 - The IUT reports the requested connection with an HCI LE Connection Complete event.
 - The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

4.3.3 INI

Tests that the IUT behaves according to the connection setup procedures as an initiator.

In all INI test cases where a connection gets created, the timing of the first packet from the master shall meet the requirements for clock accuracy. Specifically:

- Let t be the time, measured by the Lower Tester, from the end of the packet containing the CONNECT_IND or AUX_CONNECT_REQ PDU creating the connection to the start of the first data packet received from the master at the start of the connection.
- Let c be the connection event counter for that packet (normally c will equal 0, but if the Lower Tester misses the first packet from the IUT then it will be non-zero).
- Let $a = \text{transmitWindowDelay} + \text{transmitWindowOffset} + \text{connectionInterval} * c$.
- Let $d = (\text{masterSCA} + \text{slaveSCA}) / 1000000$.

Then t shall meet the requirement that:

$$a * (1 - d) - 16 \leq t \leq (a + \text{transmitWindowSize}) * (1 + d) + 16$$

The time intervals between connection events shall meet the requirements for clock accuracy.

Specifically:

- Let t_i be the time of the i -th data packet received from the master at the start of a connection event (i.e. ignoring subsequent packets in connection events).



- Let c_i be the connection event counter for that packet.
- Let $v_i = \text{connectionInterval} * ((c_i - c_{i-1}) \bmod 65536)$ be the nominal spacing between the relevant anchor points.
- Let $d_i = (\text{masterSCA} + \text{slaveSCA}) / 1000000$.

Then for every packet except the first, the timing of packet i shall meet the requirement that:

$$v_i * (1 - d_i) - 16 \leq t_i - t_{i-1} \leq v_i * (1 + d_i) + 16$$

Note: while the packets will normally be on consecutive connection events, this requirement allows for the case that one or more packets from the IUT are not received by the Lower Tester.

4.3.3.1 LL/CON/INI/BV-01-C [Connection Initiation]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when initiating.

The Lower Tester first acts in the advertising state, then accepts the connection and starts to maintain the IUT in the slave role, observing the packets and timing from the IUT.

- Reference

[3] 1.3, 4.5.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



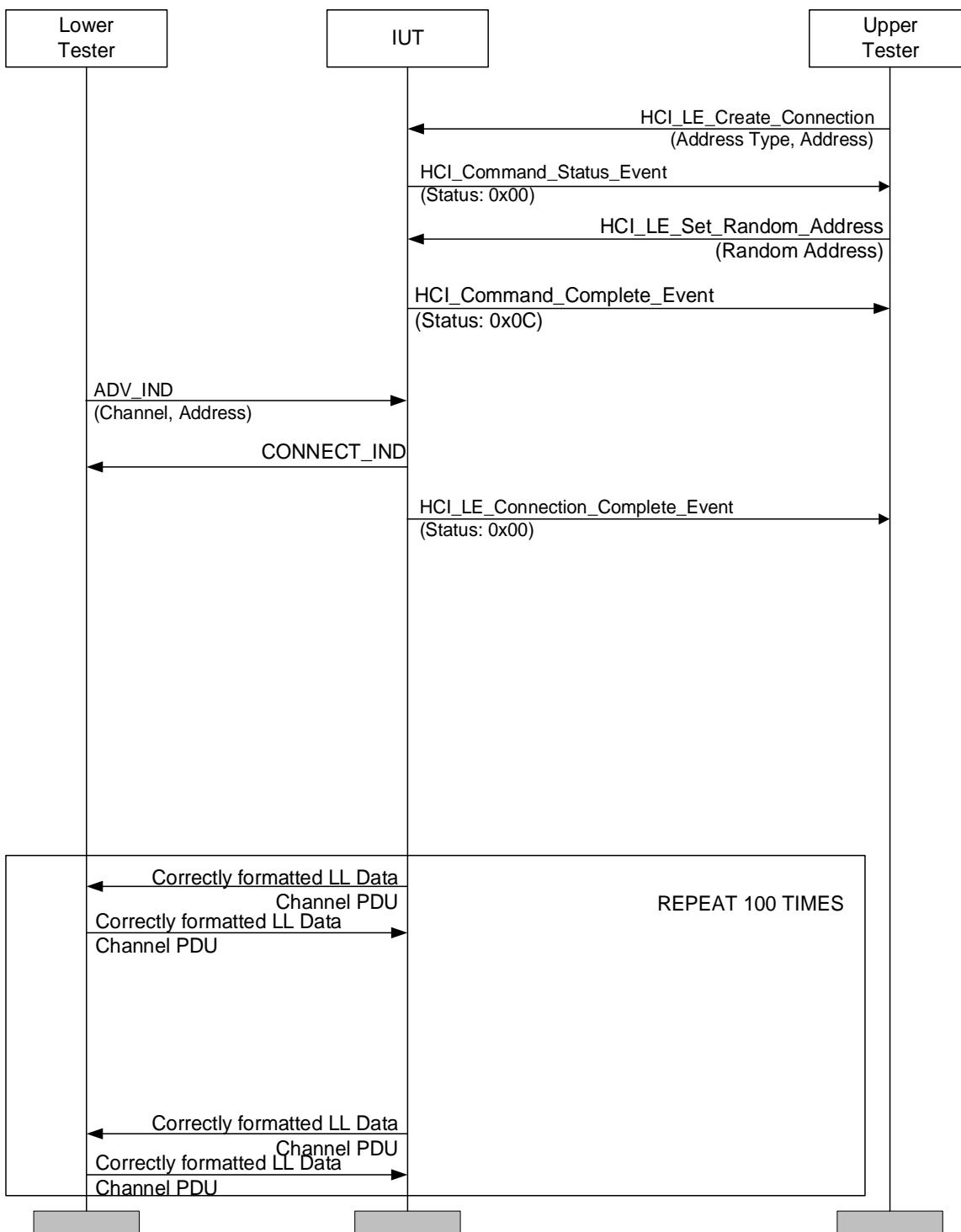


Figure 4.123: LL/CON/INI/BV-01-C [Connection Initiation]

1. Configure Lower Tester to use first supported advertising channel using a correct public address.
2. Upper Tester enables the initiator state in the IUT with the peer address and address type equal to the ones used by the Lower Tester. The Upper Tester expects an **HCI_Command_Status_Event** in response from the IUT with a Status of “success” (0x00).
3. Upper Tester sends an **HCI_LE_Set_Random_Address** command to set the IUT random address and expects an **HCI_Command_Complete** event from the IUT with a Status of 0x0C.



4. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or Step 5 executes.
5. Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_IND packets.
6. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected.
7. After the CONNECT_IND packet has been received the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
8. Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
9. Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
10. Repeat a number of events (100 events) to conclude the timing accuracy.
11. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 6).
12. Configure Lower Tester to use:
 - First supported advertising channel and an address with
 - The same company_assigned as the IUT's company_assigned setting and
 - A different company_id than the company_id setting of the IUT.
13. Repeat steps 2-11.
14. Configure Lower Tester to use
 - The first supported advertising channel and an address with
 - A different company assigned than the company_id setting of the IUT, and
 - The same company_id as the IUT's company id.
15. Repeat steps 2-11.
16. Configure Lower Tester to use
 - The first supported advertising channel, and an address with
 - A different company_assigned than company_assigned setting of the IUT, and
 - A different company_id than the company_id setting of the IUT.
17. Repeat steps 2-11.
18. Configure Lower Tester to use second supported advertising channel using a correct public address.
19. Repeat steps 2-11.
20. Configure Lower Tester to use
 - The second supported advertising channel, and an address with
 - The same company_assigned as the IUT's company_assigned setting, and
 - A different company_id than the company_id setting of the IUT.
21. Repeat steps 2-11.
22. Configure Lower Tester to use
 - The second supported advertising channel, and an address with
 - A different company_assigned than the company_assigned setting of the IUT, and
 - The same company_id as the IUT's company id.
23. Repeat steps 2-11.
24. Configure Lower Tester to use
 - The second supported advertising channel, and an address with
 - A different company_assigned than company_assigned setting of the IUT
 - A different company_id than the company_id setting of the IUT.
25. Repeat steps 2-11.
26. Configure Lower Tester to use third supported advertising channel using an address with an identical company assigned identifier but a different company identifier than the IUT.



27. Repeat steps 2-11.
28. Configure Lower Tester to use
 - The third supported advertising channel and an address with:
 - The same company_assigned as the IUT's company_assigned setting, and
 - A different company_id than the company_id setting of the IUT
29. Repeat steps 2-11.
30. Configure Lower Tester to use
 - The third supported advertising channel and an address with
 - A different company_assigned than the company_assigned setting of the IUT, and
 - The same company_id as the IUT's company id.
31. Repeat steps 2-11.
32. Configure Lower Tester to use
 - The third supported advertising channel and an address with
 - A different company_assigned than the company_assigned setting of the IUT, and
 - The same company_id as the IUT's company id.
33. Repeat steps 2-11.

- Expected Outcome

Pass Verdict

The test procedure completes using each advertising channel and device address with the IUT sending a connection request and maintaining the connection,

The IUT rejects the HCI_LE_Set_Random_Address command.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in the table below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

The IUT maintains the connection using the Channel Selection Algorithm #1.

- Notes

The state 'Connected Master', which refers to the test procedure contents above, is used as an initial state in the connection handling tests.

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms



LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
150 ppm	29.99 ms to 30.01 ms	0.45 ms
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

4.3.3.2 LL/CON/INI/BV-02-C [Connecting to Directed Advertising]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser using directed advertising events and starts to maintain a connection in the master role.

The Lower Tester first acts in the advertising state using directed advertising events, then accepts the connection and starts to maintain it in the slave role.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



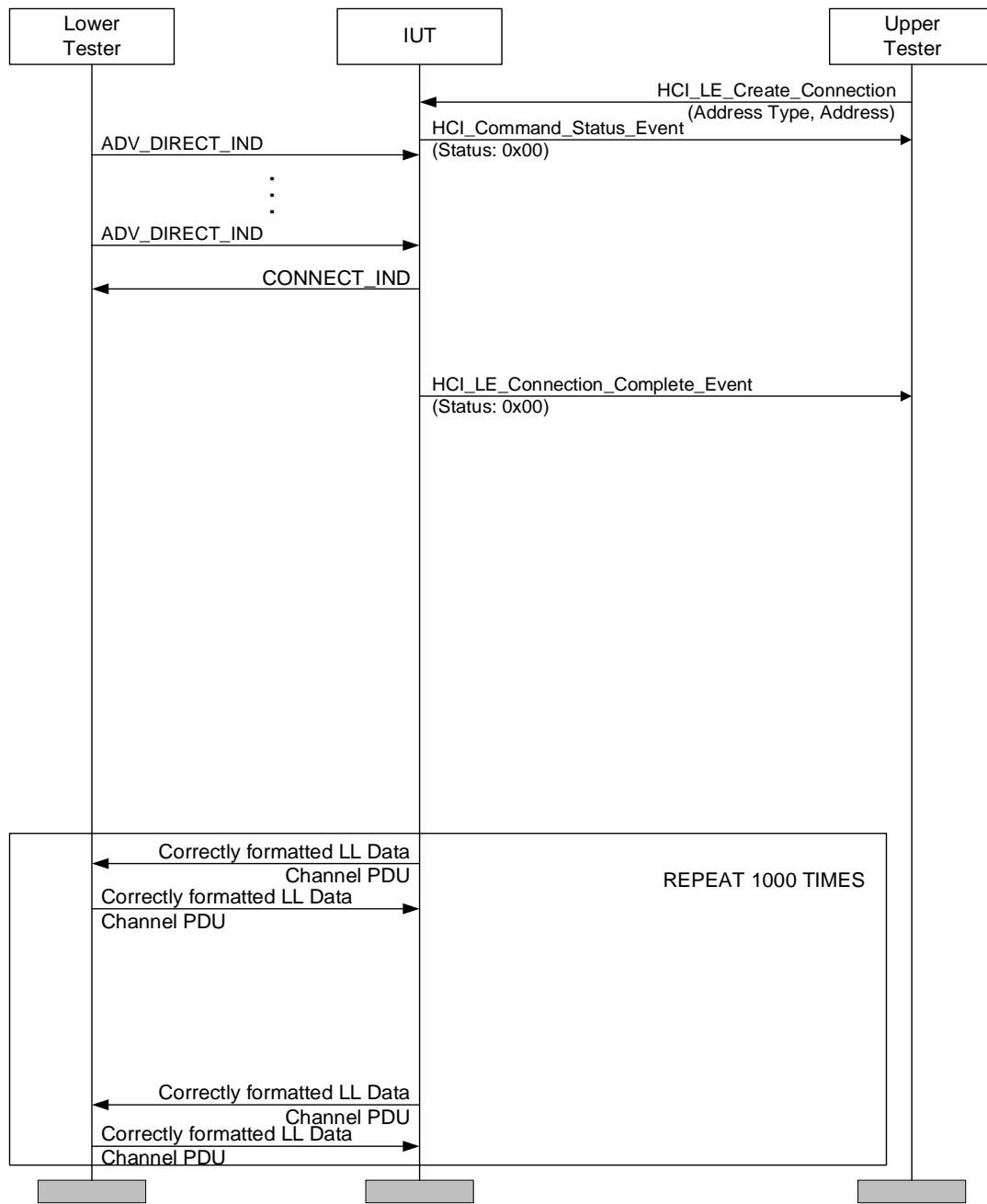


Figure 4.124: LL/CON/INI/BV-02-C [Connecting to Directed Advertising]

1. Configure Lower Tester as advertiser using all supported advertising channel with the interval between packets at most 0.5 ms and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends ADV_DIRECT_IND packets, each advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or step 4 executes.
4. Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_DIRECT_IND packets.
5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.



6. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
 7. Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 8. Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 9. Repeat a number of events (at least 1000 events) to conclude the timing accuracy.
 10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 5).
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection,

The connection events' time intervals are within the range expressed for the sleep clock accuracy value,

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

The IUT maintains the connection using the Channel Selection Algorithm #1.

4.3.3.3 LL/CON/INI/BV-03-C [Connection Initiation Missed Replies]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and after missing some reply transmissions from the slave, still manages to setup a connection in the master role. This test case reflects a typical scenario which the IUT must manage.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, and then begins to maintain the connection after omitting some events above the slave latency figure.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window and using the common device address. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



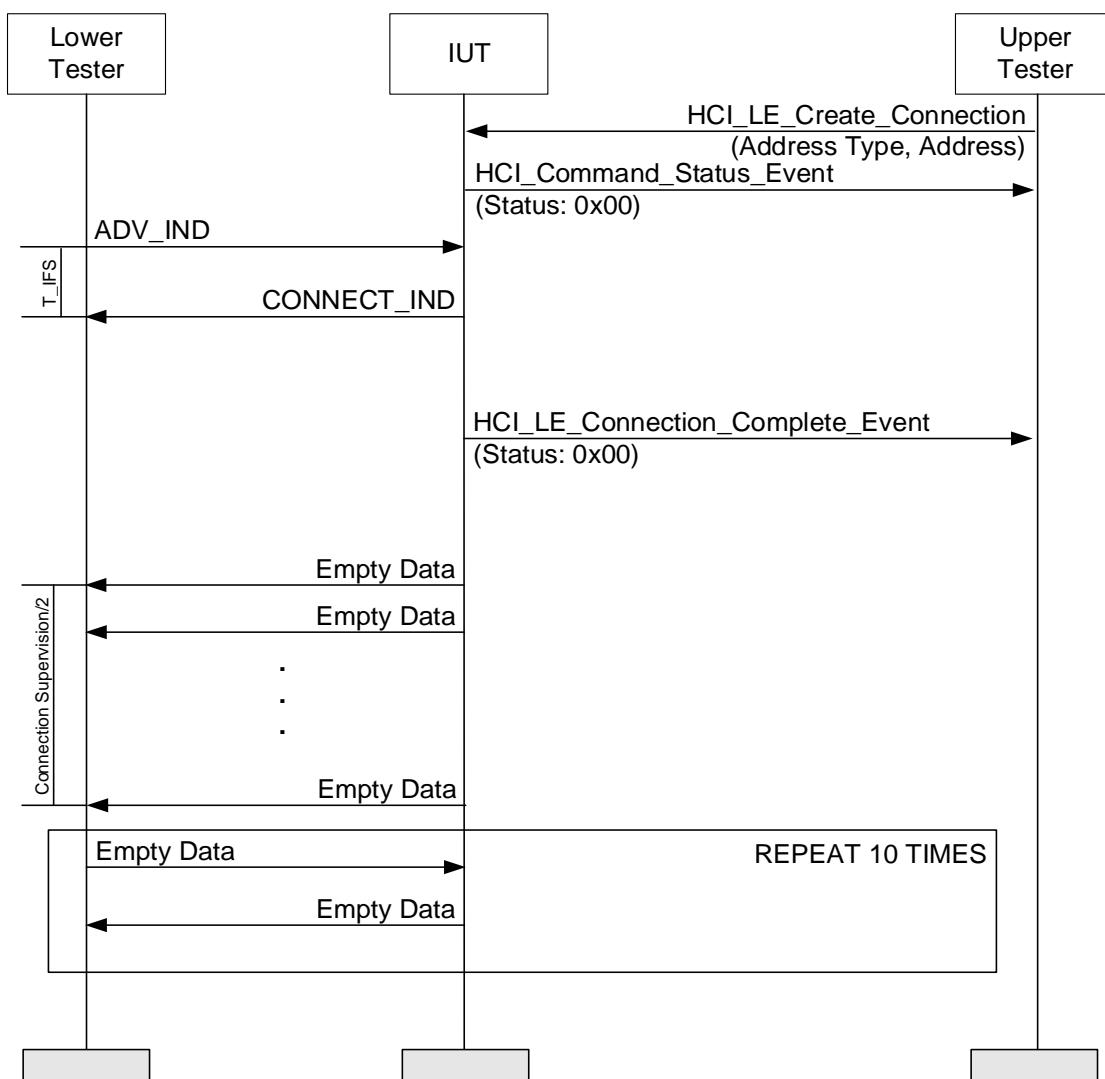


Figure 4.125: LL/CON/INI/BV-03-C [Connection Initiation Missed Replies]

1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends **ADV_IND** packets, each advertising event on the selected advertising channel only using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or Step 4 executes.
4. Lower Tester expects a **CONNECT_IND** packet **T_IFS** after any of the **ADV_IND** packets.
5. Upper Tester expects an **HCI_LE_Connection_Complete_Event** from the IUT including the Lower Tester address and connection interval selected.
6. After the **CONNECT_IND** packet has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel in the range of maximum/minimum deviation of the allowed **transmitWindowOffset** and **transmitWindowSize**.
7. Lower Tester does not reply to the correctly formatted LL Data Channel PDU and expects following correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals of minus (latest plus) maximum clock drift according to the drift rate indicated in the connection request, calculated for the connection interval used. Repeat for half the time required for the connection supervision timer to expire, then stop.

8. Lower Tester expects a correctly formatted LL Data Channel PDU and sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 9. Repeat a number of events (at least 10 events).
 10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 5).
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request, then continuing master transmissions until and after receiving the slave transmissions,

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses.

4.3.3.4 LL/CON/INI/BV-04-C [Connection Initiation Timeout]

- Test Purpose

Tests that an initiator IUT, after sending a connection request to an advertiser and missing reply transmissions from the slave until the connection supervision timer expires, considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, then does not start to maintain the connection, but observes the IUT reports in HCI events.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, 0 slave latency, 100 ms timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval, to maintain the connection. Use a zero slave latency AND a timeout parameter of 100 ms.



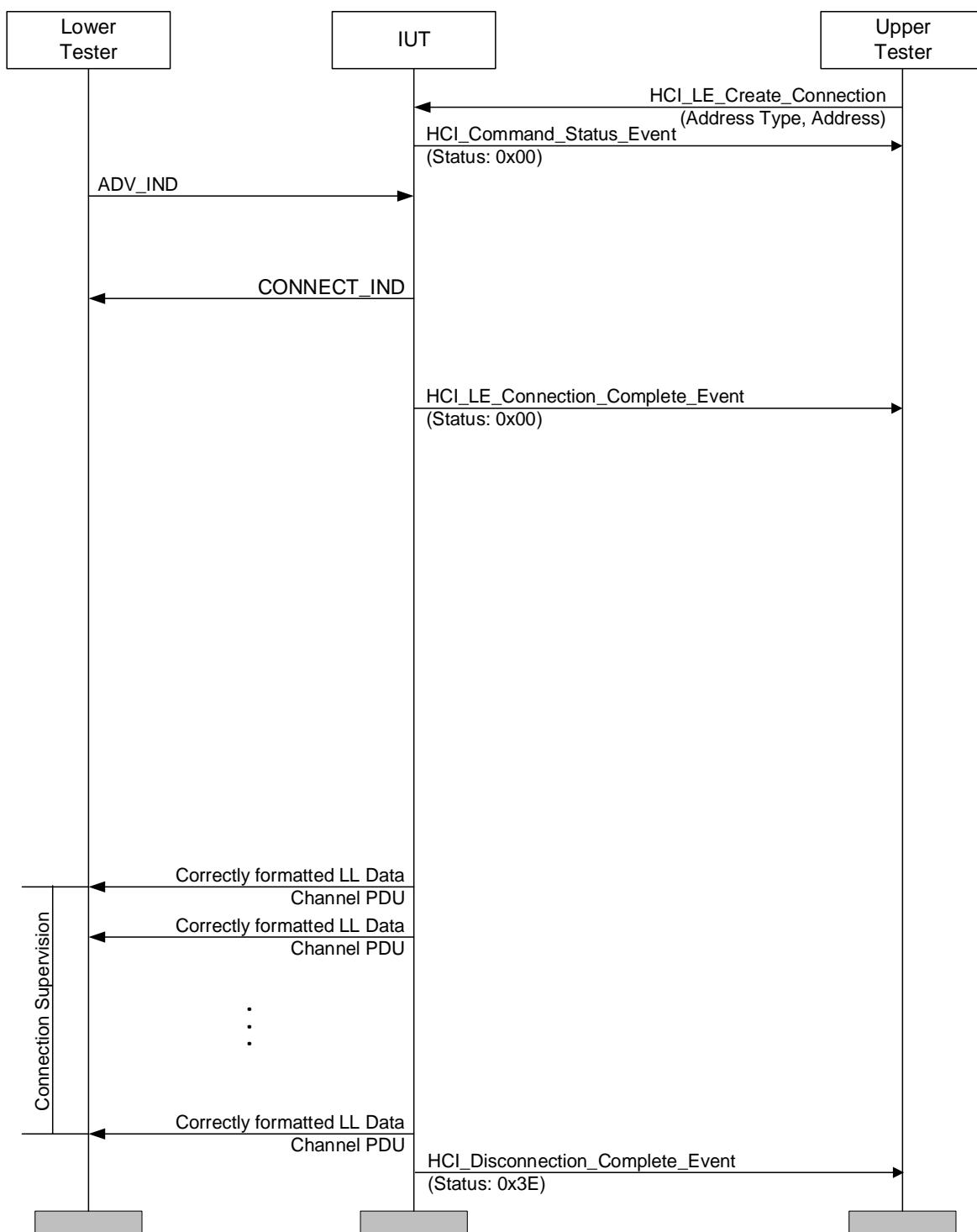


Figure 4.126: LL/CON/INI/BV-04-C [Connection Initiation Timeout]

1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiator state in the IUT.
3. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel only using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or Step 4 executes.
4. Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_IND packets.

5. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 6. After the CONNECT_IND packet has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel, in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
 7. Lower Tester does not reply to the packet and expects correctly formatted LL Data Channel PDUs on subsequent data channels.
 8. Repeat step 7 until the IUT stops.
 9. Upper Tester expects an HCI_Disconnection_Complete event from the IUT with the reason parameter set to 'connection failed to be established' and with the connection handle matching to Step 5.
- Expected Outcome
- Pass Verdict
- The IUT transmits a connection request,
- The IUT then continues master transmissions until the connection supervision timer expires,
- The IUT reports failure to establish a connection with an HCI event.

4.3.3.5 LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected]

- Test Purpose
- Tests that an initiator IUT sends connection requests correctly filtering advertiser devices.
- The Lower Tester acts in the advertising state using undirected advertising events, observing the connection request packets from the IUT.
- Reference
- [3] 4.5.4, 4.3.4
- Initial Condition
- Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
 LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
 LL_initiator_Channel_Map,
- State: Initiating (selected scan interval, selected scan window, white list is used, selected type of peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout) AND Connection Setup White Listed (Lower Tester address, selected type of peer address)



- Test Procedure

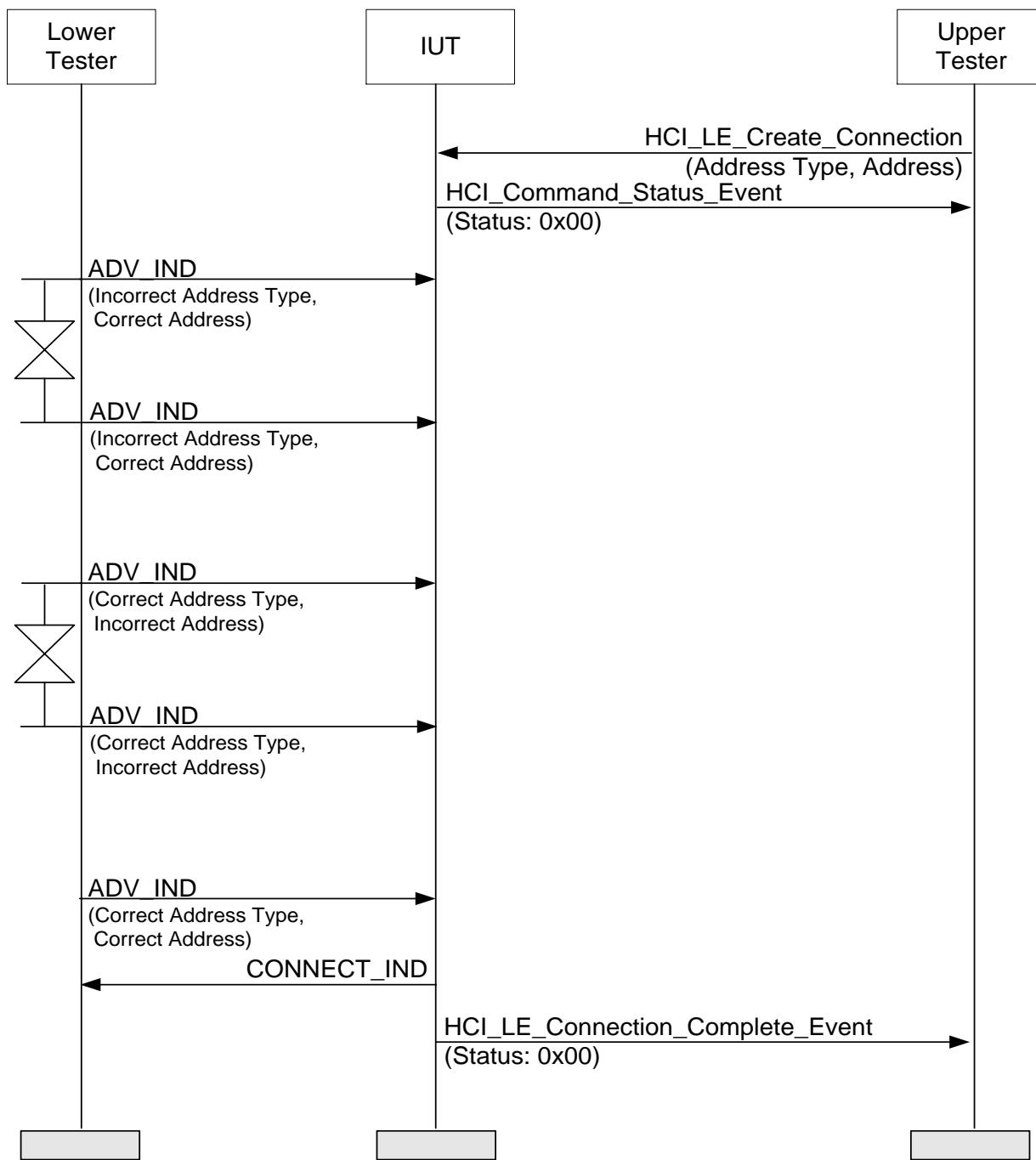


Figure 4.127: LL/CON/INI/BV-06-C [Initiation Device Filtering: Undirected]

- Configure the White List of the IUT: public address type and a common device address
- Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a white listed device address and a random address type.
- Upper Tester enables initiator state in the IUT using a public address type and the white listed device address.

4. Lower Tester sends ADV_IND packets each advertising event, using as advertising data the event sequence numbering and using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$.
 5. Lower Tester expects no CONNECT_IND after any of the ADV_IND packets.
 6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type.
 7. Repeat steps 4-5.
 8. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
 9. Lower Tester sends ADV_IND packets, with event count as data encoded unsigned least significant bit first. Lower Tester sends the ADV_IND packet search advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or Step 13 executes.
 10. Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_IND packets.
 11. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 12. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).
 13. Repeat steps 1-12 changing public address type to random address type.
- Expected Outcome

Pass Verdict

The IUT sends a connection request to a white listed advertiser with the address of the initiator in the advertising packet,

The IUT does not send a connection request to a black listed advertiser with the address of the initiator in the advertising packet,

The IUT reports the connection requested with an HCI event.

The access address used by the IUT meets the requirements for access addresses.

4.3.3.6 LL/CON/INI/BV-07-C [Initiation Device Filtering: Directed]

- Test Purpose

Tests that an initiator IUT sends connection requests correctly filtering advertiser devices.

The Lower Tester acts in the advertising state using directed advertising events, observing the connection request packets from the IUT.

- Reference

[3] 4.5.4, 4.3.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
 LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
 LL_initiator_Channel_Map,

State: Initiating (selected scan interval, selected scan window, white list is used, selected type of peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout) AND Connection Setup White Listed (Lower Tester address, selected type of peer address)



- Test Procedure

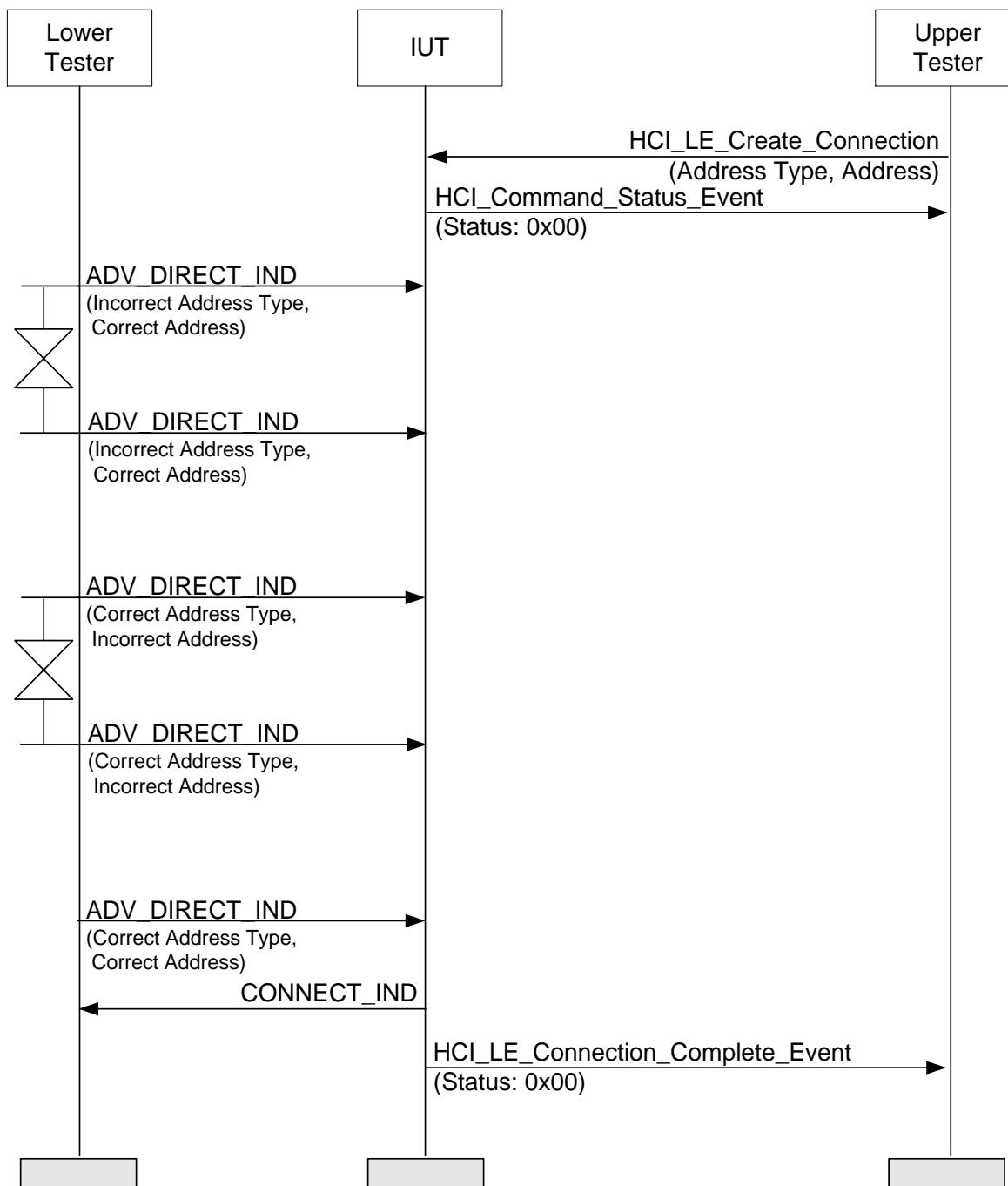


Figure 4.128: LL/CON/INI/BV-07-C [Initiation Device Filtering: Directed]

- Configure the White List of the IUT: public address type and a common device address
- Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a white listed device address and a random address type.
- Upper Tester enables initiator state in the IUT using a public address type and the white listed device address.
- Lower Tester sends ADV_DIRECT_IND packets each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$.



5. Lower Tester expects no CONNECT_IND after any of the ADV_DIRECT_IND packets.
 6. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), a device address other than the white listed device address and a public address type.
 7. Repeat steps 4-5.
 8. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), the white listed device address and a public address type.
 9. Lower Tester sends ADV_DIRECT_IND packets, each advertising event using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or Step 13 executes.
 10. Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packets.
 11. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 12. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).
 13. Repeat steps 1-12 changing public address type to random address type.
- Expected Outcome

Pass Verdict

The IUT does not respond to connection requests from the Lower Tester in steps 4 through 8.

The IUT sends a connection request to the white listed Lower Tester advertiser with the correct address of the initiator in the advertising packet,

The IUT reports the connection requested with an HCI event.

4.3.3.7 LL/CON/INI/BV-08-C [Network Privacy – Connection Establishment responding to connectable undirected advertising, Initiator]

- Test Purpose

Verify that the IUT, when initiating connection establishment with the resolving list only containing a local IRK, the IUT uses the AdvA field received from the Lower Tester in the connectable undirected advertising event and generates a resolvable private address for the InitA field in the connect request packet.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



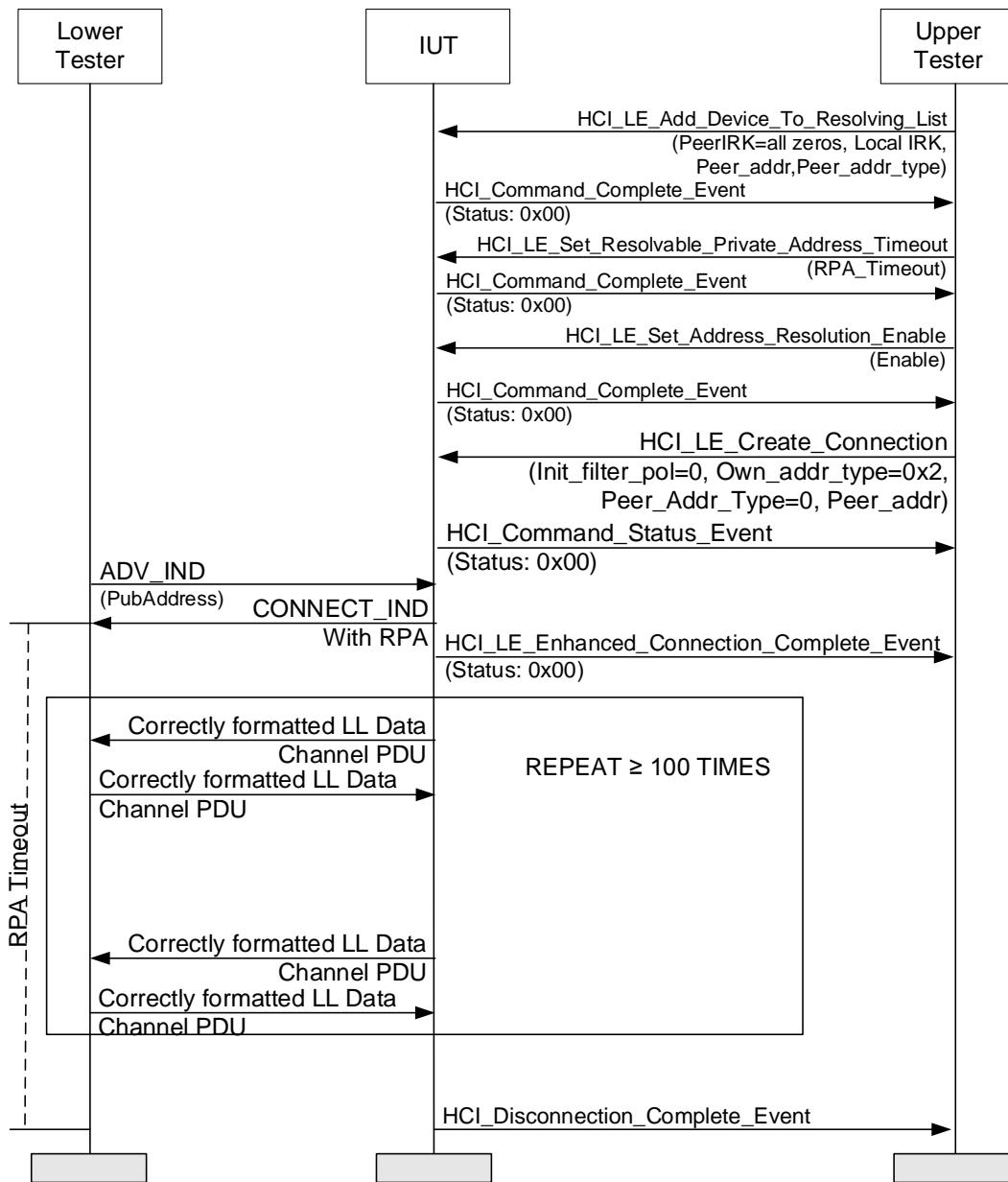


Figure 4.129: LL/CON/INI/BV-08-C [Network Privacy – Connection Establishment responding to connectable undirected advertising, Initiator]

1. Configure the Lower Tester to advertise using a valid public or static random address (identity address).
 2. The Upper Tester adds the Identity Address of the Lower Tester to the Resolving List with all zero peer IRK, and with a local IRK.
 3. The Upper Tester enables the initiator state in the IUT.
 4. Lower Tester sends ADV_IND packets, each advertising event on the selected advertising channel, using the selected advertising interval.
 5. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_IND packets. The InitA field contains a resolvable private address from the IUT.
 6. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 7. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.

8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request.

The IUT reports the connection setup with the HCI event.

The InitA field in the connection request uses a properly generated resolvable private address.

4.3.3.8 LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator]

- Test Purpose

Verify that the IUT when initiating connection establishment only connects to devices that are in the resolving list. The Lower Tester uses connectable undirected advertising.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using a resolvable private address in the AdvA field of the advertising packets.

The Lower Tester has previously distributed its IRK to the IUT.



- Test Procedure

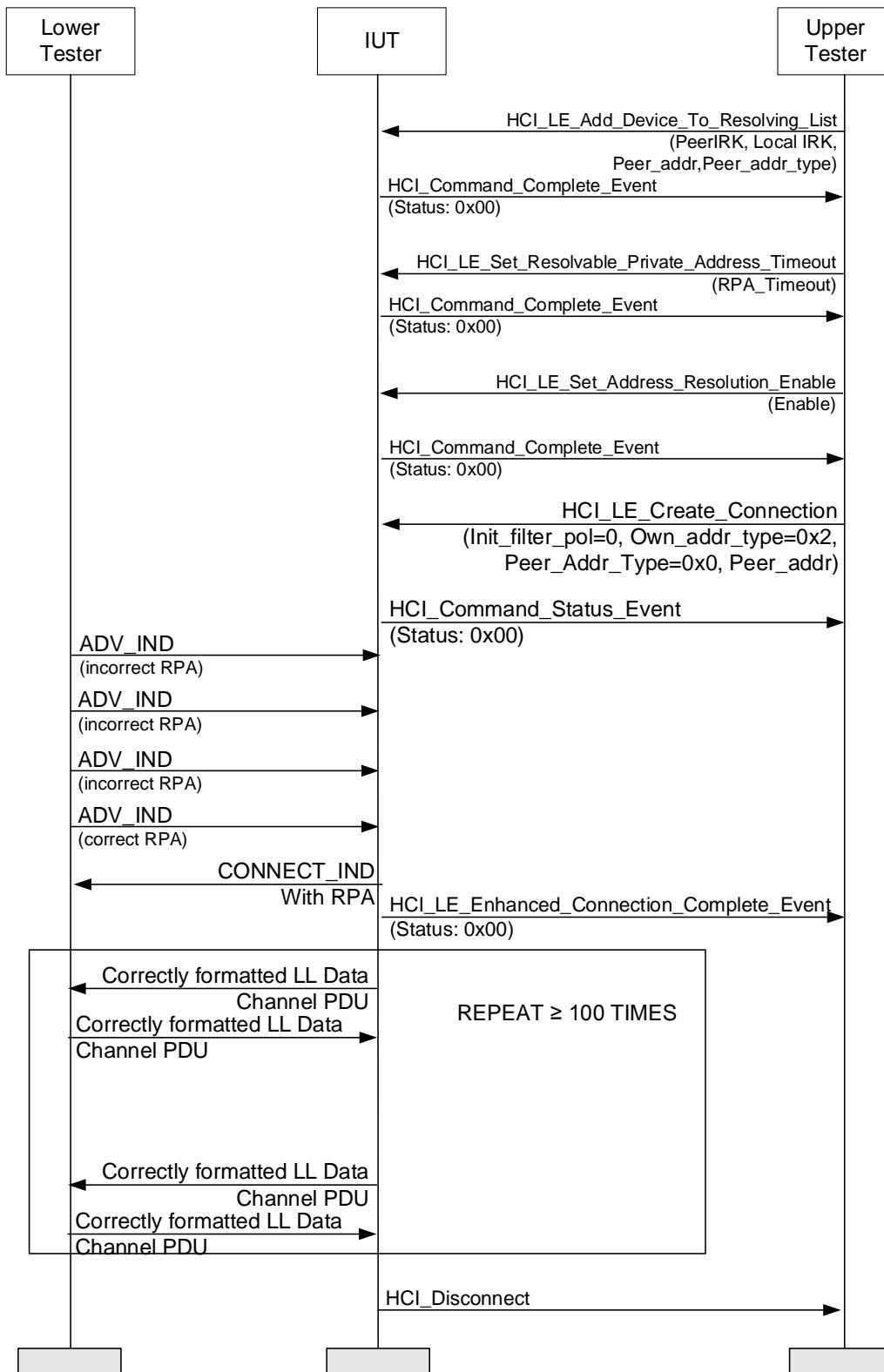


Figure 4.130: LL/CON/INI/BV-09-C [Network Privacy – Connection Establishment using resolving list, Initiator]

1. Configure the Lower Tester to start advertising with a resolvable private address generated from a random IRK.

2. The Upper Tester adds the Lower Tester to the resolving list using a different IRK than in step 1.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$.
5. The IUT compares the address by checking against its resolving list and does not find a match.
6. The Lower Tester expects no CONNECT_IND after any of the ADV_IND packets.
7. The Lower Tester stops advertising.
8. The Lower Tester begins advertising again using the correct resolvable address, which matches the one in the IUT resolving list. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$, or step 9 occurs.
9. The Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
10. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester's RPA and Identity address and connection interval selected.
11. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
13. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
15. The Upper Tester terminates the connection.

- Expected Outcome

Pass Verdict

The IUT receives and ignores connectable advertising with a resolvable private address which is not in the resolving list.

The IUT then receives a resolvable private address in the AdvA field from the Lower Tester which is in the resolving list.

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The IUT reports the connection setup with an HCI event.

4.3.3.9 LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator]

- Test Purpose

Verify that the IUT when initiating connection establishment with the resolving list connects only to peer devices that are in the resolving list. The Lower Tester uses directed advertising. The IUT may use a public or static random device address.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map



State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

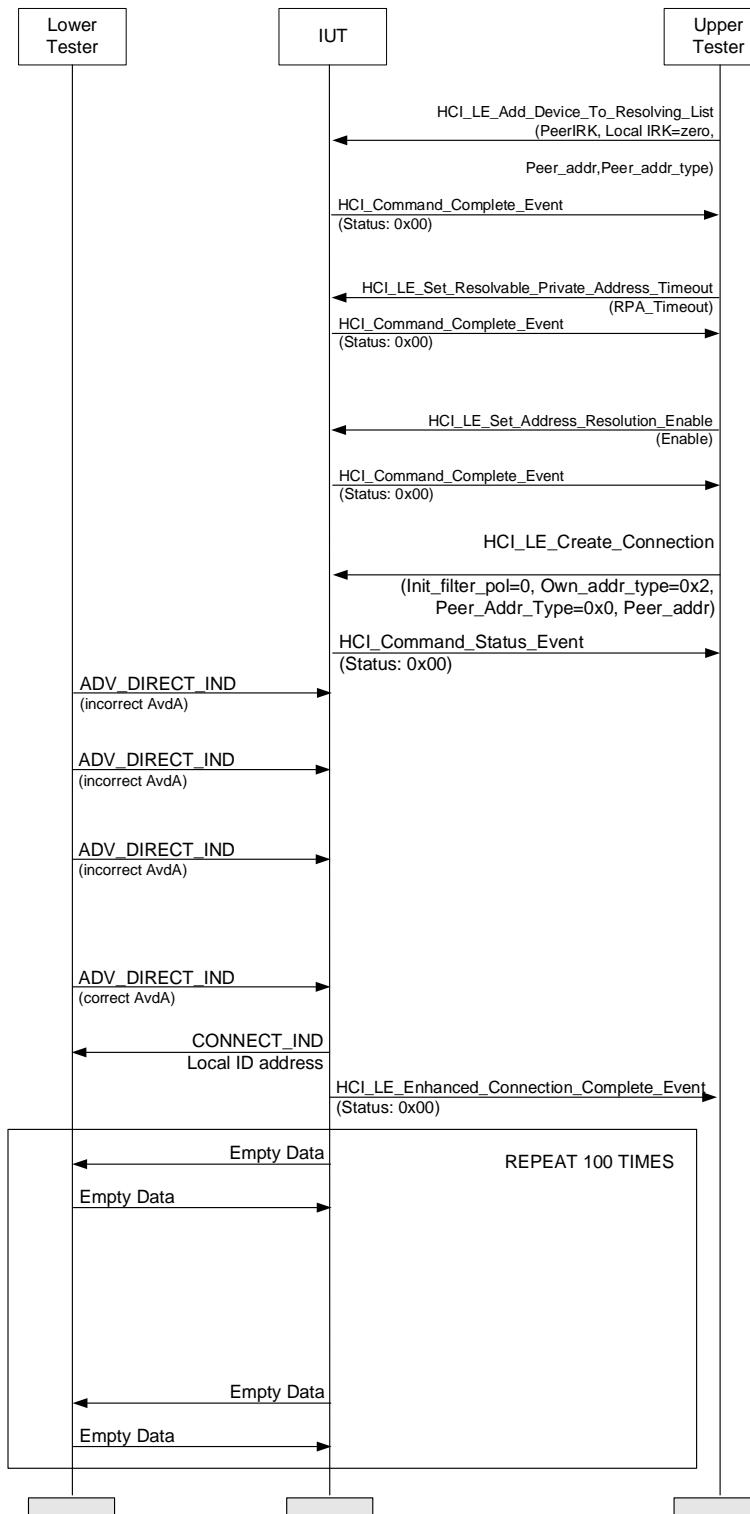


Figure 4.131: LL/CON/INI/BV-10-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator]



1. Configure the Lower Tester to use the first supported advertising channel and a valid resolvable private address in the AdvA field, which is not known by the IUT (not in the resolving list).
 2. The Upper Tester adds the Lower Tester to the resolving list using a different IRK than in step 1.
 3. The Upper Tester enables the initiator state in the IUT.
 4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.
 5. The IUT resolves and compares the address in the AdvA field by checking against its resolving list and does not find a match.
 6. Repeat the advertising steps 4-5 for at least 20 advertising intervals.
 7. The Lower Tester stops advertising.
 8. The Lower Tester begins advertising again using the correct resolvable address, which matches the one stored in the IUT's resolving list.
 9. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_DIRECT_IND packets. The AdvA field is the same as the one received in the ADV_DIRECT_IND.
 10. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 11. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 13. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
 15. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

The IUT receives directed advertising with a resolvable private address in the AdvA field from the Lower Tester.

The first AdvA address used does not match the resolving list and is ignored.

The 2nd AdvA address used matches the resolving list and is accepted.

The IUT uses the identity address in the InitA field of the connect request packet.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

4.3.3.10 LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator]

- Test Purpose

Verify that the IUT when initiating connection establishment with the resolving list connects only to directed advertisements that are addressed to the IUT.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
 LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
 LL_initiator_Channel_Map



State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

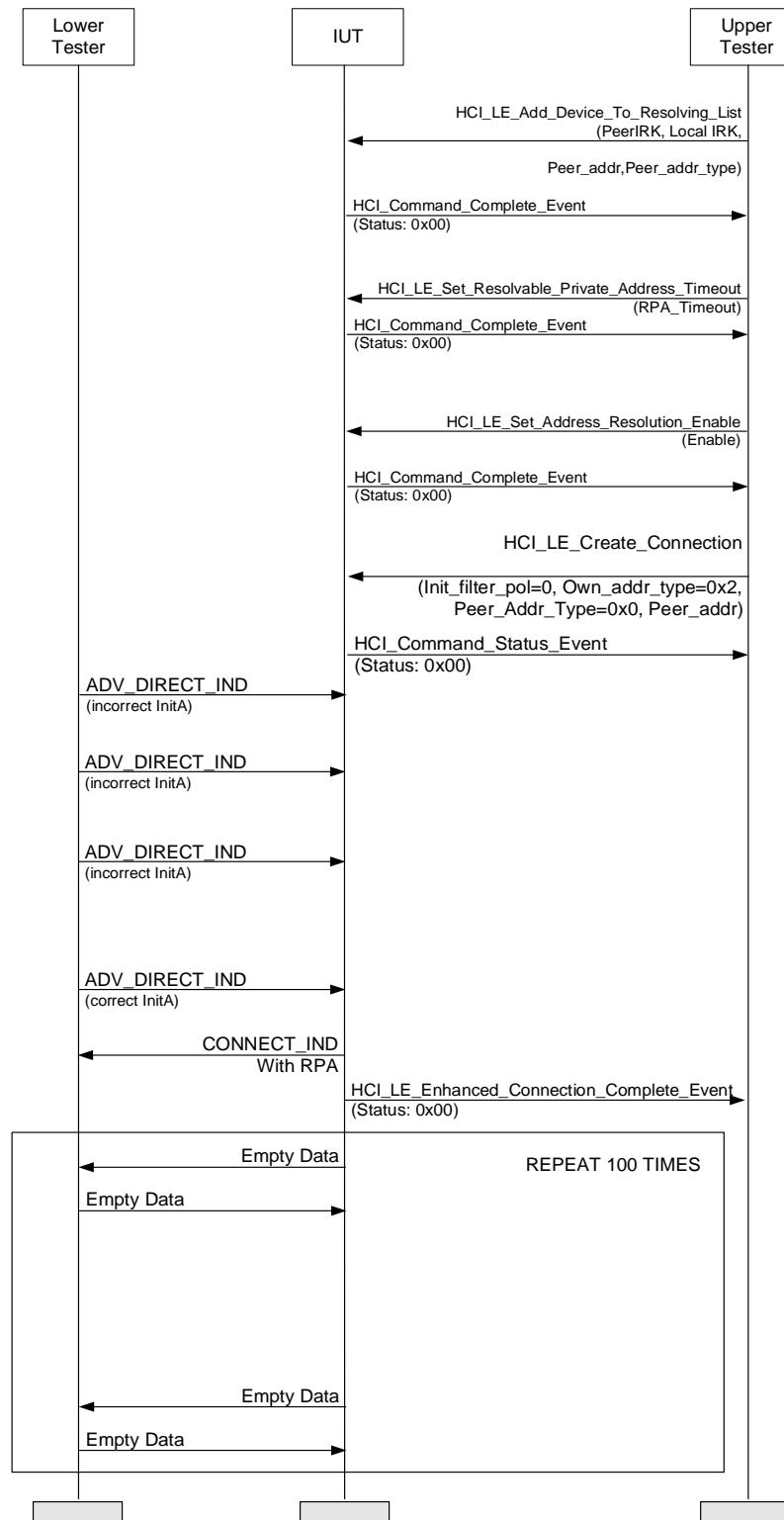


Figure 4.132: LL/CON/INI/BV-11-C [Network Privacy – Connection Establishment using directed advertising with wrong address and resolving list, Initiator]



1. Configure the Lower Tester to start directed advertising using a valid resolvable private address in the AdvA field.
 2. The Upper Tester adds the Lower Tester to the resolving list with both peer Identity (IRK and Identity Address) and local IRK.
 3. The Upper Tester enables the initiator state in the IUT.
 4. Lower Tester sends ADV_DIRECT_IND packets, each advertising event, using the selected advertising interval and with a resolvable private address in the InitA field generated from a random IRK different from the one distributed to the IUT.
 5. The IUT tries to resolve the address in the InitA field by checking against its resolving list and does not find a match.
 6. Repeat the advertising steps 3-5 for at least 20 advertising intervals.
 7. The Lower Tester stops advertising.
 8. The Lower Tester begins advertising again using the correct resolvable address, which resolves with the IUT's local IRK.
 9. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_DIRECT_IND packets. The InitA field contains a resolvable private address generated by the IUT. The address should be different from the address received in the ADV_DIRECT_IND packet.
 10. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 11. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 12. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 13. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 14. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
 15. The IUT (master) maintains the link for the address refresh timeout before terminating the connection.
 16. Repeat steps 3-10 to see that the InitA field of the CONNECT_IND have been refreshed.
 17. The Upper Tester terminates the connection.
- Expected Outcome
- Pass Verdict
- The IUT receives directed advertising with a resolvable private address in the AdvA field and an invalid resolvable private address in the InitA field from the Lower Tester.
- The first InitA address used does not match the resolving list and is ignored.
- The second InitA address used matches the resolving list and is accepted.
- The IUT sends a resolvable private address in the InitA field of the connect request packet.
- The IUT address is verified against the resolving list.
- The Upper Tester then disconnects after the address refresh timeout has passed and reconnects to see that the IUT generates a new resolvable private address.
- The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.



4.3.3.11 LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator]

- Test Purpose

Verify the IUT when initiating private connection establishment with the resolving list does not connect to directed advertisements that are addressed to the IUT using its identity address.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)



- Test Procedure

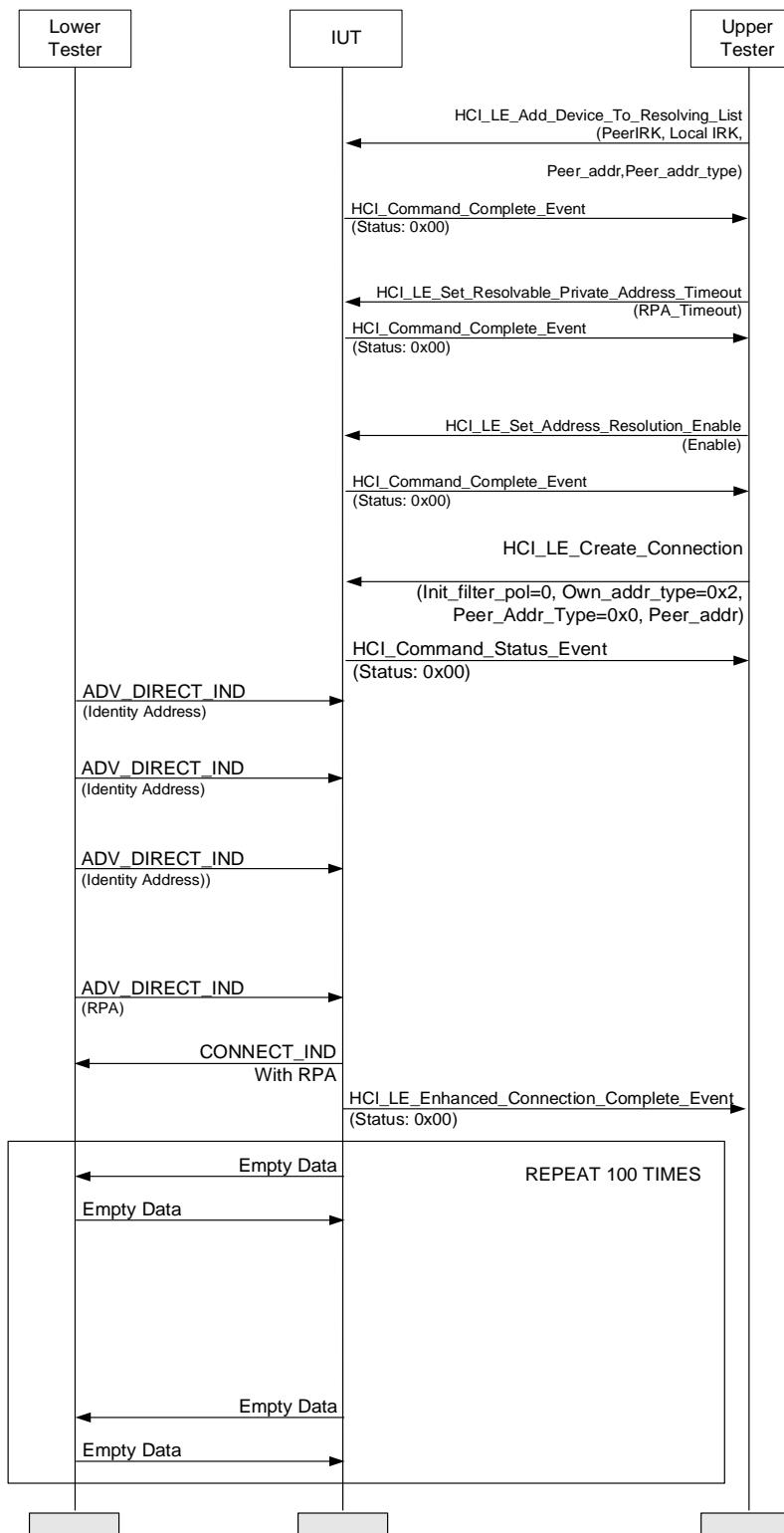


Figure 4.133: LL/CON/INI/BV-12-C [Network Privacy – Connection Establishment using directed advertising with identity address and resolving list, Initiator]

1. The Upper Tester adds the address of the Lower Tester to the resolving list with the peer IRK.
2. The Upper Tester enables the initiator state in the IUT using a private address.



3. Configure the Lower Tester to start directed advertising using a valid resolvable private address in the AdvA field and the identity address of the IUT (either the public address or the static random address configured in the initial conditions).
 4. Lower Tester sends ADV_DIRECT_IND packets, each advertising event, using the selected advertising interval and with public address of the IUT in the InitA.
 5. The IUT sees its own public address in the ADV_DIRECT_IND, but ignores it since it is initiating using a private address.
 6. Repeat steps 4-5 for at least 20 advertising intervals.
 7. The Lower Tester starts advertising again using a resolvable private address generated using the IUT IRK in the InitA field.
 8. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_DIRECT_IND packets. The InitA field contains a resolvable private address generated by the IUT. The address should be different from the address received in the ADV_DIRECT_IND packet.
 9. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 10. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 11. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 12. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 13. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
 14. The IUT (master) maintains the link for the address refresh timeout before terminating the connection
 15. Repeat steps 3-9 to see that the InitA field of the CONNECT_IND have been refreshed.
 16. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

The IUT receives directed advertising with a resolvable private address in the AdvA field and the identity address in the InitA field from the Lower Tester.

The first InitA address used does not match the own_address_type to use only Resolvable Private Address and is ignored.

The second InitA address used matches the resolving list and is accepted.

The IUT sends a resolvable private address in the InitA field of the connect request.

The IUT address is verified against the resolving list.

The Upper Tester then disconnects after the address refresh timeout has passed and reconnects to see that the IUT generates a new resolvable private address.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

4.3.3.12 LL/CON/INI/BV-13-C [Extended Scanning, Connection Initiation]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser on the secondary advertising channel and starts to maintain a connection in the master role. The Lower Tester first acts in the extended advertising state, then accepts the connection and starts to maintain the IUT in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel



Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference
[\[10\]](#) 4.3.4, 4.4.4, 4.5.3, 4.5.4
- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



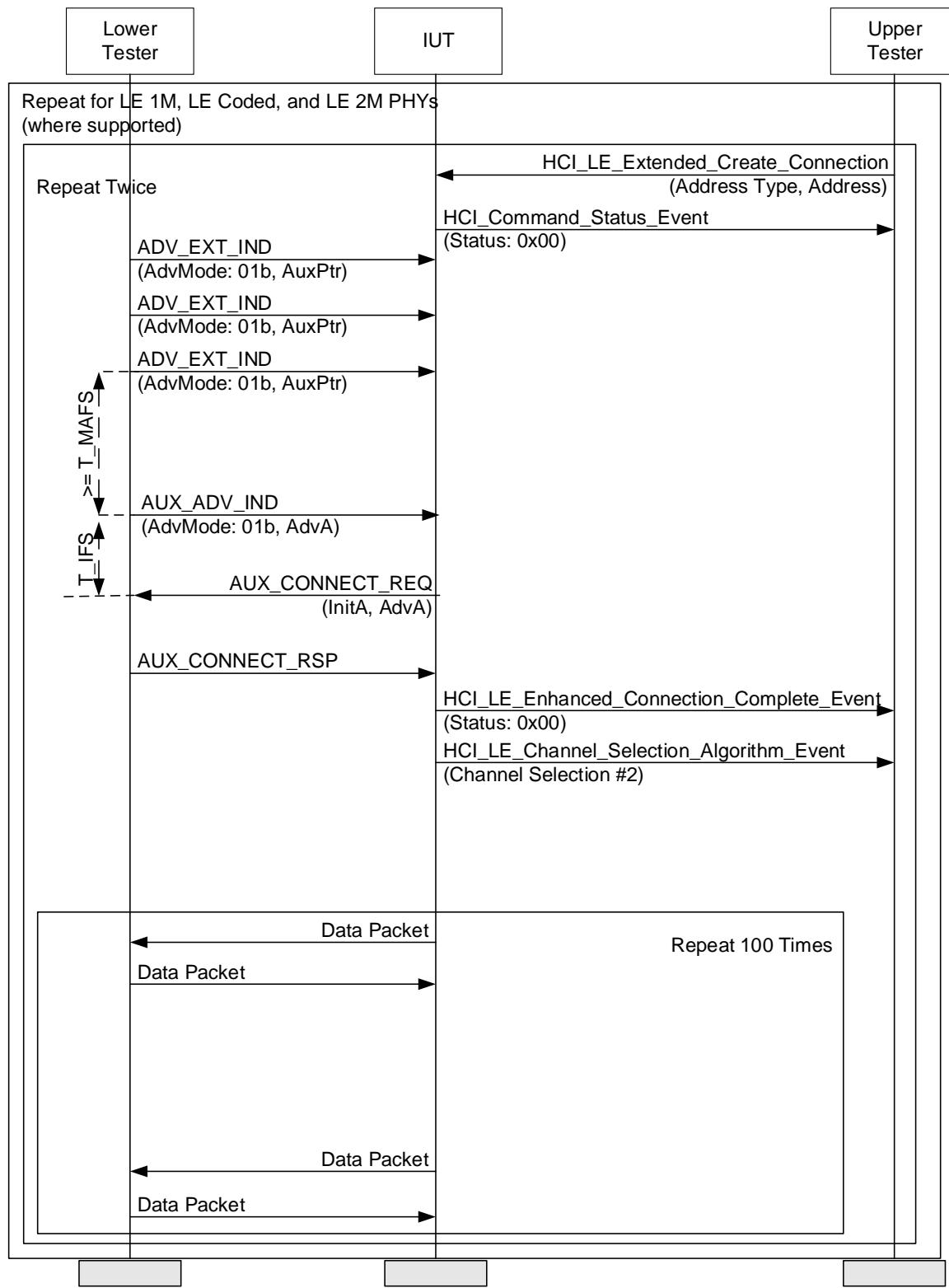


Figure 4.134: LL/CON/INI/BV-13-C [Extended Scanning, Connection Initiation]

1. The Upper Tester sends an **HCI_LE_Extended_Create_Connection** command to the IUT. The **Initiating_PHYs** parameter shall be set to 0x01 (LE 1M), **Scan_Interval[0]** set to 0x0010, and **Scan_Window[0]** set to 0x0010. **Initiating_Filter_Policy** shall be set to 0x00 (use **Peer_Address_Type** and **Peer_Address**), and **Own_Address_Type** shall be set to 0x00 (Public



- Device Address). The peer address and address type shall be set to the ones used by the Lower Tester. The Upper Tester expects an HCI_Command_Status event in response.
2. The Lower Tester begins advertising using the ADV_EXT_IND PDU on the LE 1M PHY with the AuxPtr field referencing the AUX_ADV_IND.
 3. The Lower Tester expects an AUX_CONNECT_REQ PDU on the secondary advertising channel on the LE 1M PHY T_IFS after sending any of the AUX_ADV_IND PDUs.
 4. The Lower Tester sends an AUX_CONNECT_RSP PDU from the IUT on the secondary advertising channel T_IFS after receiving the AUX_CONNECT_REQ.
 5. The Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x01 (LE Channel Selection Algorithm #2 is used).
 6. After the AUX_CONNECT_RSP has been sent, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from AUX_CONNECT_REQ.
 7. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
 8. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 9. Repeat a number of events (100 events) to conclude the timing accuracy.
 10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 5).
 11. Repeat Steps 1-10, except instead of Step 4, the Lower Tester does not respond to the first AUX_CONNECT_REQ PDU, continues advertising, and expects the IUT to retry the connection request following the backoff algorithm by sending a second AUX_CONNECT_REQ PDU on the secondary advertising channel on the LE 1M PHY T_IFS after any of the AUX_ADV_IND PDUs.
 12. If the IUT supports the LE Coded PHY, repeat steps 1-11, except that in Step 1 the Initiating_PHYs shall be set to 0x04 (LE Coded) and in Steps 2, 3, 4, and 11, the Lower Tester uses the LE Coded PHY for both primary and secondary advertising channels and expects the IUT to respond using the LE Coded PHY for the secondary advertising channel.
 13. If the IUT supports the LE 2M PHY, repeat steps 1-11, except that in Step 1 the Initiating_PHYs shall be set to 0x01 (LE 1M) and in Steps 2, 3, 4, and 11, the Lower Tester uses the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel and expects the IUT to respond using the LE 2M PHY for the secondary advertising channel.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.28](#) below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.



The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

When a connection request receives no response, the IUT retries according to the backoff algorithm.

The access address used by the IUT meets the requirements for access addresses

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms
150 ppm	29.99 ms to 30.01 ms	0.45 ms
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

Table 4.28: Timing requirements

4.3.3.13 LL/CON/INI/BV-14-C [Connection Initiation, Channel Selection Algorithm #1]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates no support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to zero (0), then accepts the connection and starts to maintain it in the slave role, observing the packet and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #1 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)



- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.

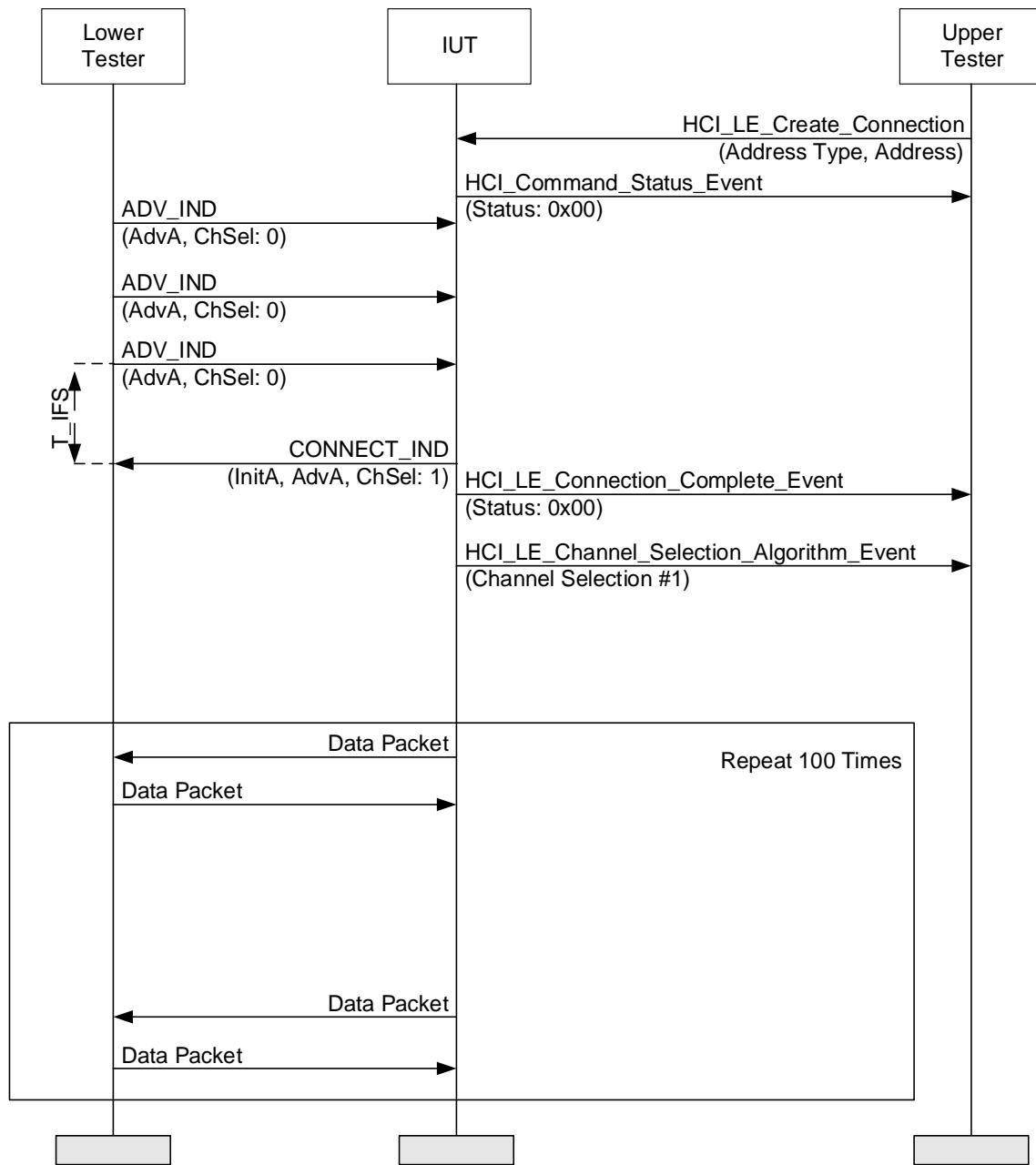


Figure 4.135: LL/CON/INI/BV-14-C [Connection Initiation, Channel Selection Algorithm #1]

1. The Upper Tester sends an **HCI_LE_Create_Connection** command, with the peer address and address type equal to the ones used by the Lower Tester and the **Initiator_Filter_Policy** set to 0x00 (use **Peer_Address_Type** and **Peer_Address**). The Upper Tester expects an **HCI_Command_Status** event in response.
2. The Lower Tester begins advertising using **ADV_IND** PDUs with ChSel set to 0.
3. The Lower Tester expects a **CONNECT_IND** PDU with ChSel set to 1 T_IFS after sending any of the **ADV_IND** PDUs.

4. The Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester expects an HCI_LE_Channel_Selection_Algorithm event from the IUT with Channel_Selection_Algorithm set to 0x00 (LE Channel Selection Algorithm #1 is used).
5. The Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
7. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from Step 4).

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.29](#) below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms
150 ppm	29.99 ms to 30.01 ms	0.45 ms



LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

Table 4.29: Timing requirements.

4.3.3.14 LL/CON/INI/BV-15-C [Connecting to Directed Advertising, Channel Selection Algorithm #1]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser using directed advertising and starts to maintain a connection in the master role when the advertisement indicates no support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to zero (0), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #1 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



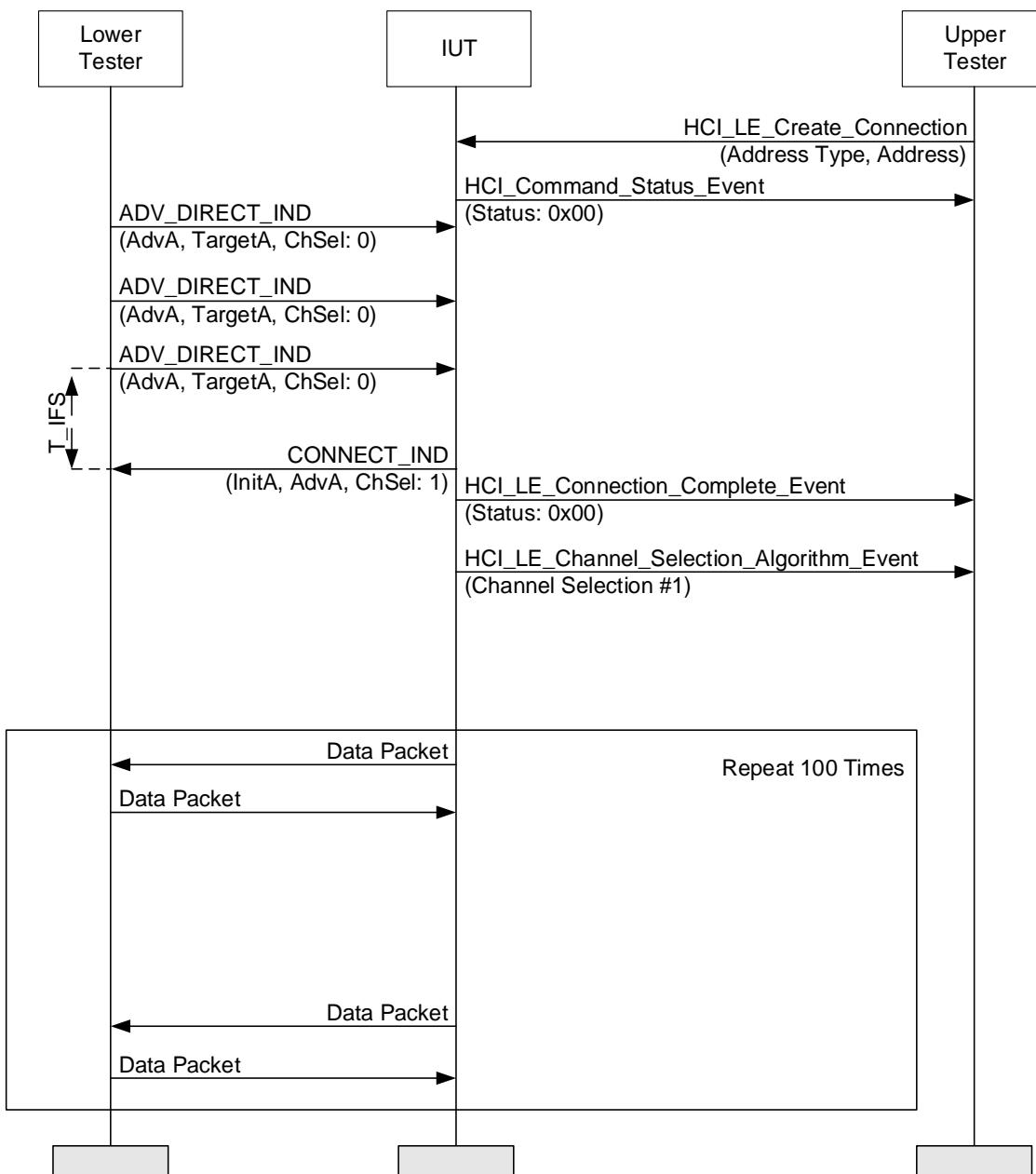


Figure 4.136: LL/CON/INI/BV-15-C [Connecting to Directed Advertising, Channel Selection Algorithm #1]

1. The Upper Tester sends an **HCI_LE_Create_Connection** command, with the peer address and address type equal to the ones used by the Lower Tester and the **Initiator_Filter_Policy** set to 0x00 (use **Peer_Address_Type** and **Peer_Address**). The Upper Tester expects an **HCI_Command_Status** event in response.
2. The Lower Tester begins advertising using **ADV_DIRECT_IND** PDUs with **ChSel** set to 0.
3. The Lower Tester expects a **CONNECT_IND** PDU with **ChSel** set to 1 T_IFS after sending any of the **ADV_DIRECT_IND** PDUs.
4. The Upper Tester expects an **HCI LE Connection Complete** event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester expects an **HCI LE Channel Selection Algorithm** event from the IUT with **Channel_Selection_Algorithm** set to 0x00 (LE Channel Selection Algorithm #1 is used).

5. The Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
7. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from Step 4).

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.30](#) below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms
150 ppm	29.99 ms to 30.01 ms	0.45 ms
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms



LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

Table 4.30: Timing requirements.

4.3.3.15 LL/CON/INI/BV-16-C [Connection Initiation, Channel Selection Algorithm #2]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



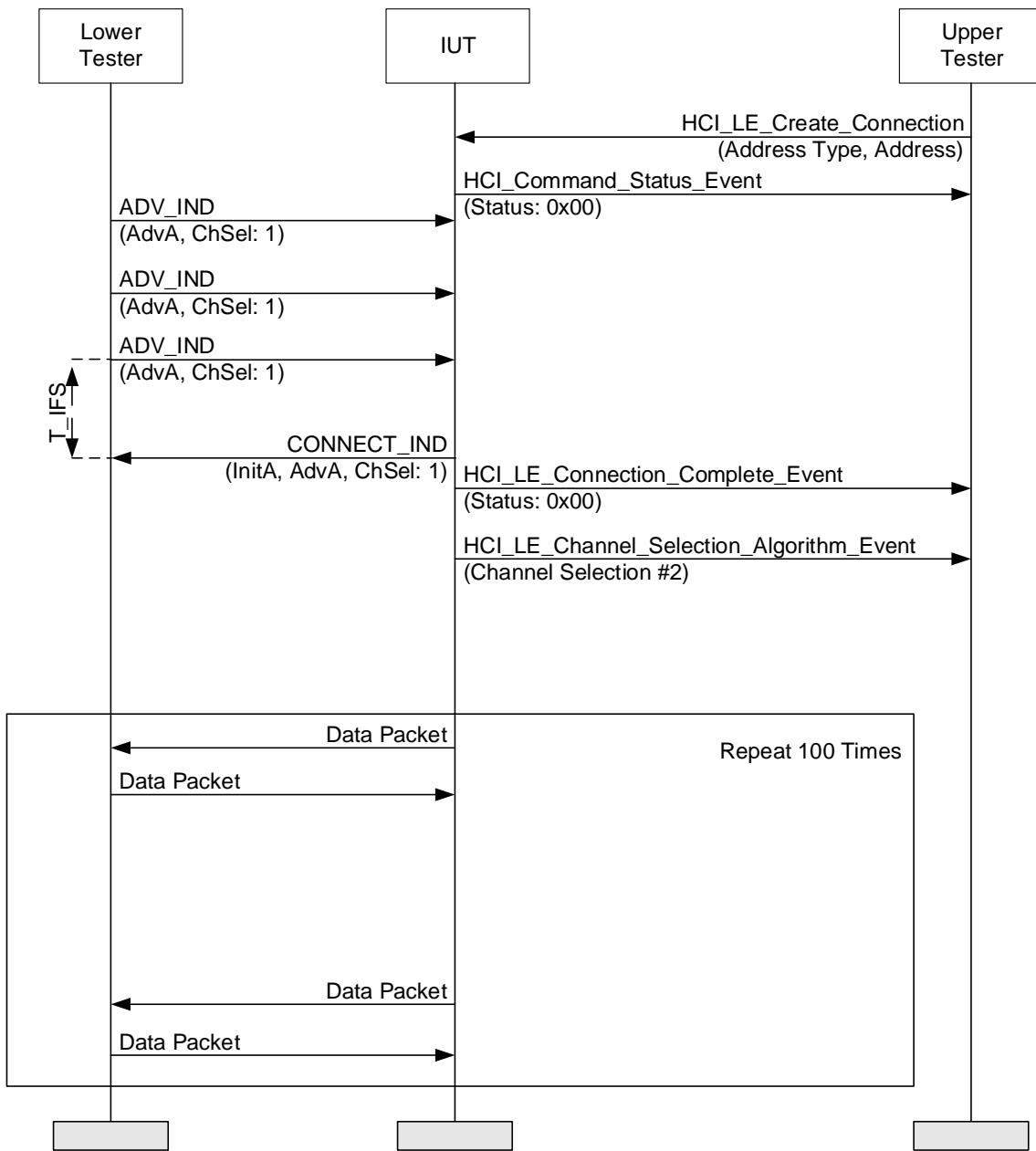


Figure 4.137: LL/CON/INI/BV-16-C [Connection Initiation, Channel Selection Algorithm #2]

1. The Upper Tester sends an **HCI_LE_Create_Connection** command, with the peer address and address type equal to the ones used by the Lower Tester and the **Initiator_Filter_Policy** set to 0x00 (use **Peer_Address_Type** and **Peer_Address**). The Upper Tester expects an **HCI_Command_Status** event in response.
2. The Lower Tester begins advertising using **ADV_IND** PDUs with **ChSel** set to 1.
3. The Lower Tester expects a **CONNECT_IND** PDU with **ChSel** set to 1 T_{IFS} after sending any of the **ADV_IND** PDUs.
4. The Upper Tester expects an **HCI_LE_Connection_Complete** event from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester expects an **HCI_LE_Channel_Selection_Algorithm** event from the IUT with **Channel_Selection_Algorithm** set to 0x01 (LE Channel Selection Algorithm #2 is used).

5. The Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
7. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
8. Repeat a number of events (100 events) to conclude the timing accuracy.
9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.31](#) below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses.

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms
150 ppm	29.99 ms to 30.01 ms	0.45 ms
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms



LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

Table 4.31: Timing requirements.

4.3.3.16 LL/CON/INI/BV-17-C [Connecting to Directed Advertising, Channel Selection Algorithm #2]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser using directed advertising and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packets and timing from the IUT. The Lower Tester confirms the Channel Selection Algorithm #2 is used for the connection. The Upper Tester confirms the LE Channel Selection Algorithm Event is generated.

- Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.



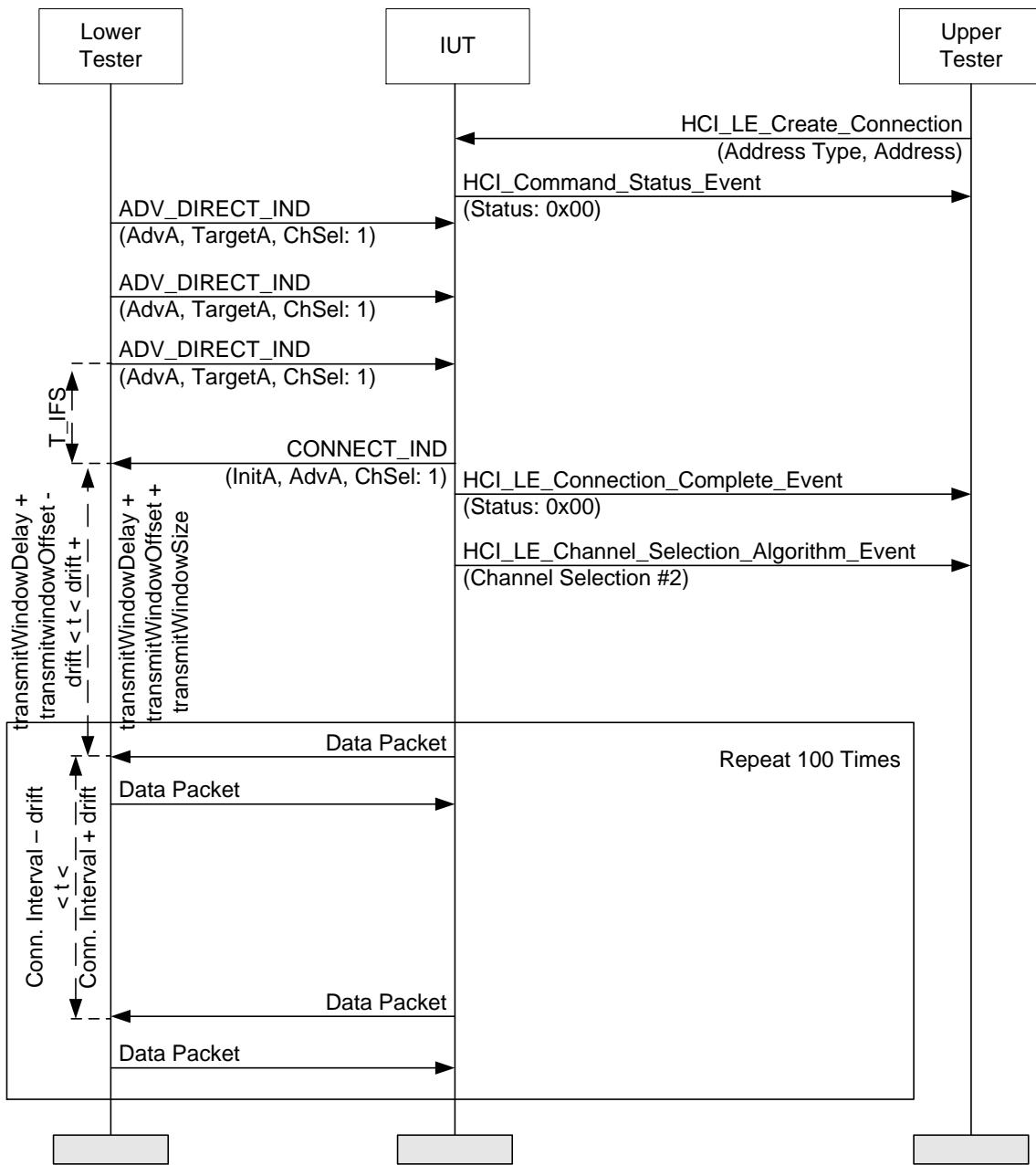


Figure 4.138: LL/CON/INI/BV-17-C [Connecting to Directed Advertising, Channel Selection Algorithm #2]

1. The Upper Tester sends an **HCI_LE_Create_Connection** command, with the peer address and address type equal to the ones used by the Lower Tester and the **Initiator_Filter_Policy** set to 0x00 (use **Peer_Address_Type** and **Peer_Address**). The Upper Tester expects an **HCI_Command_Status_Event** in response.
2. The Lower Tester begins advertising using **ADV_DIRECT_IND** PDUs with ChSel set to 1.
3. The Lower Tester expects a **CONNECT_IND** PDU with ChSel set to 1 T_{IFS} after sending any of the **ADV_DIRECT_IND** PDUs.
4. The Upper Tester expects an **HCI_LE_Connection_Complete_Event** from the IUT including the Lower Tester address and the connection interval selected. Immediately after, the Upper Tester expects an **HCI_LE_Channel_Selection_Algorithm_Event** from the IUT with **Channel_Selection_Algorithm** set to 0x01 (LE Channel Selection Algorithm #2 is used).

5. The Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #2 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
 6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
 7. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 8. Repeat a number of events (100 events) to conclude the timing accuracy.
 9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.32](#) below.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around T_IFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The IUT reports the channel selection algorithm used for the connection with an HCI LE Channel Selection Algorithm Event.

The access address used by the IUT meets the requirements for access addresses

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #2.

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted:

LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
500 ppm	29.98 ms to 30.02 ms	1.50 ms
250 ppm	29.99 ms to 30.01 ms	0.75 ms
150 ppm	29.99 ms to 30.01 ms	0.45 ms
100 ppm	29.99 ms to 30.01 ms	0.30 ms
75 ppm	29.99 ms to 30.01 ms	0.23 ms
50 ppm	29.99 ms to 30.01 ms	0.15 ms



LL_SCA	Event time interval accepted	Limit for drift accepted / 100 intervals
30 ppm	29.99 ms to 30.01 ms	0.09 ms
20 ppm	29.99 ms to 30.01 ms	0.06 ms

Table 4.32: Timing requirements.

4.3.3.17 LL/CON/INI/BV-18-C [Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address]

- Test Purpose

Verify that the IUT when initiating connection establishment does not connect to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.

The Lower Tester has previously distributed its IRK to the IUT.

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.



- Test Procedure

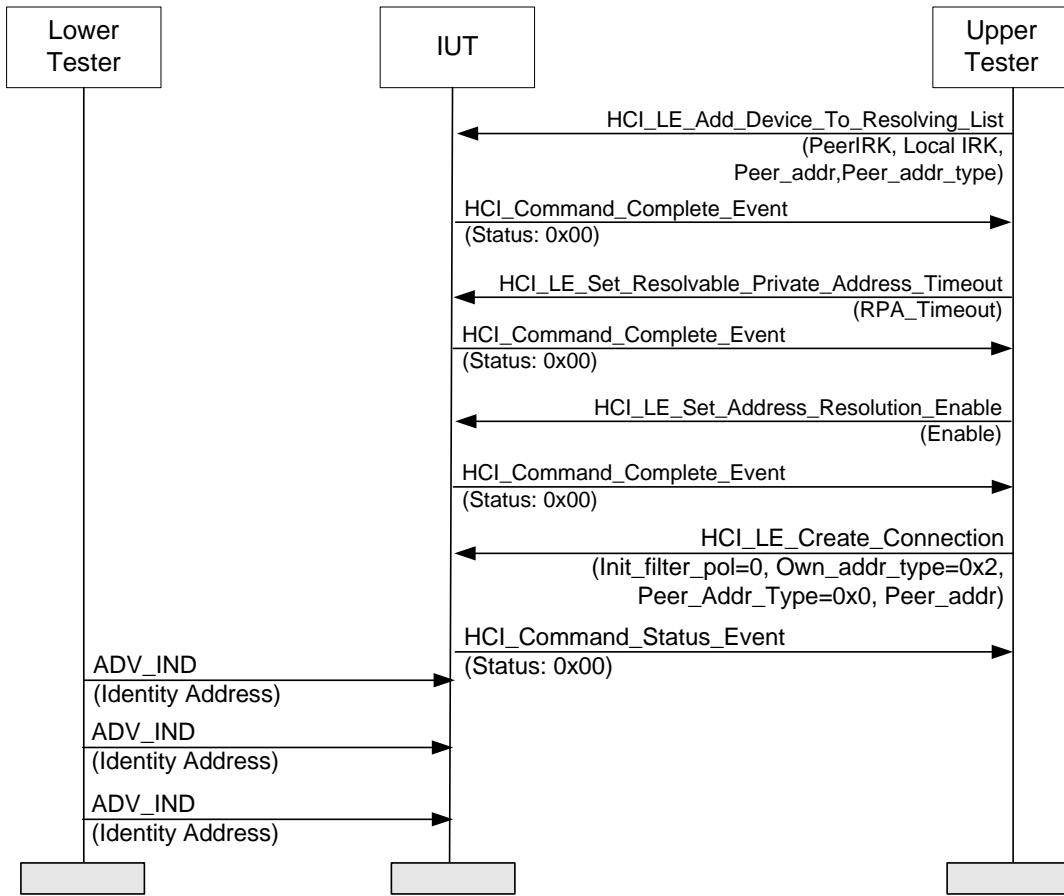


Figure 4.139: LL/CON/INI/BV-18-C [Network Privacy – Connection Establishment using resolving list, Initiator, Ignore Identity Address]

1. Configure the Lower Tester to start advertising with its device identity address.
 2. The Upper Tester adds the Lower Tester to the resolving list using an IRK.
 3. The Upper Tester enables the initiator state in the IUT.
 4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$.
 5. The IUT compares the address by checking against its resolving list and finds a match with network privacy mode.
 6. The Lower Tester expects no CONNECT_IND after any of the ADV_IND packets.
 7. The Lower Tester stops advertising.

Expected Outcome

Pass Verdict
The IUT receives and ignores connectable advertising with a device identity address when the Identity Address is not present in the MAC address of the advertising device.

4.3.3.18 LL/CON/INI/BV-19-C [Network Privacy – Connection Establishment using directed advertising on Association list initiator, Impr. Identity Address]

- 1 —

Verify that the IUT when initiating connection establishment does not connect to a device advertising its local identity and has local identity and has no associate LIDK and the machine

list using network privacy mode. The Lower Tester uses directed advertising. The IUT may use a public or static random device address.

- Reference
- [3] 1.3, 4.5.4
- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.

- Test Procedure

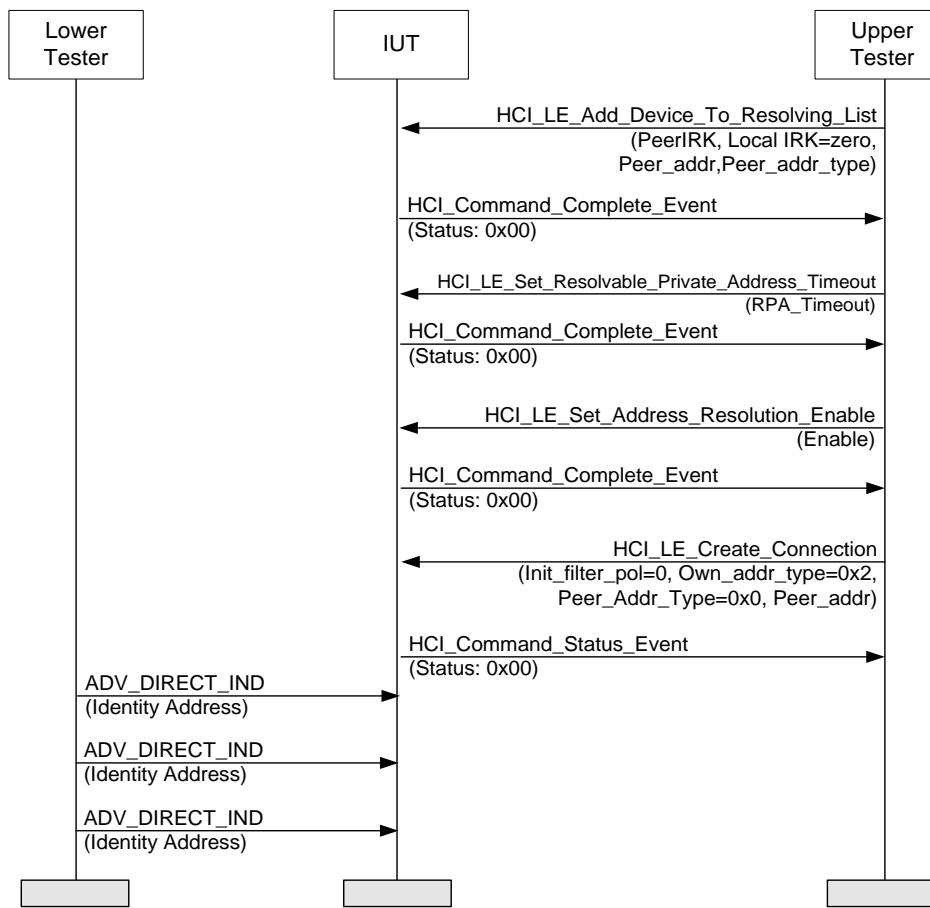


Figure 4.140: LL/CON/INI/BV-19-C [Network Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Ignore Identity Address]

1. Configure the Lower Tester to start directed advertising with its device identity address.
2. The Upper Tester adds the Lower Tester to the resolving list using an IRK.
3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.



5. The IUT resolves and compares the address in the AdvA field by checking against its resolving list and finds a match with network privacy mode.
 6. The Lower Tester expects no CONNECT_IND after any of the ADV_DIRECT_IND packets.
 7. Repeat the advertising steps 4-5 for at least 20 advertising intervals.
 8. The Lower Tester stops advertising.
- Expected Outcome

Pass Verdict

The IUT receives and ignores connectable advertising with a device identity address when the identity address and an associated IRK are in the resolving list using network privacy mode.

4.3.3.19 LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address]

- Test Purpose

Verify that the IUT when initiating connection establishment connects to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.

The Lower Tester has previously distributed its IRK to the IUT.



- Test Procedure

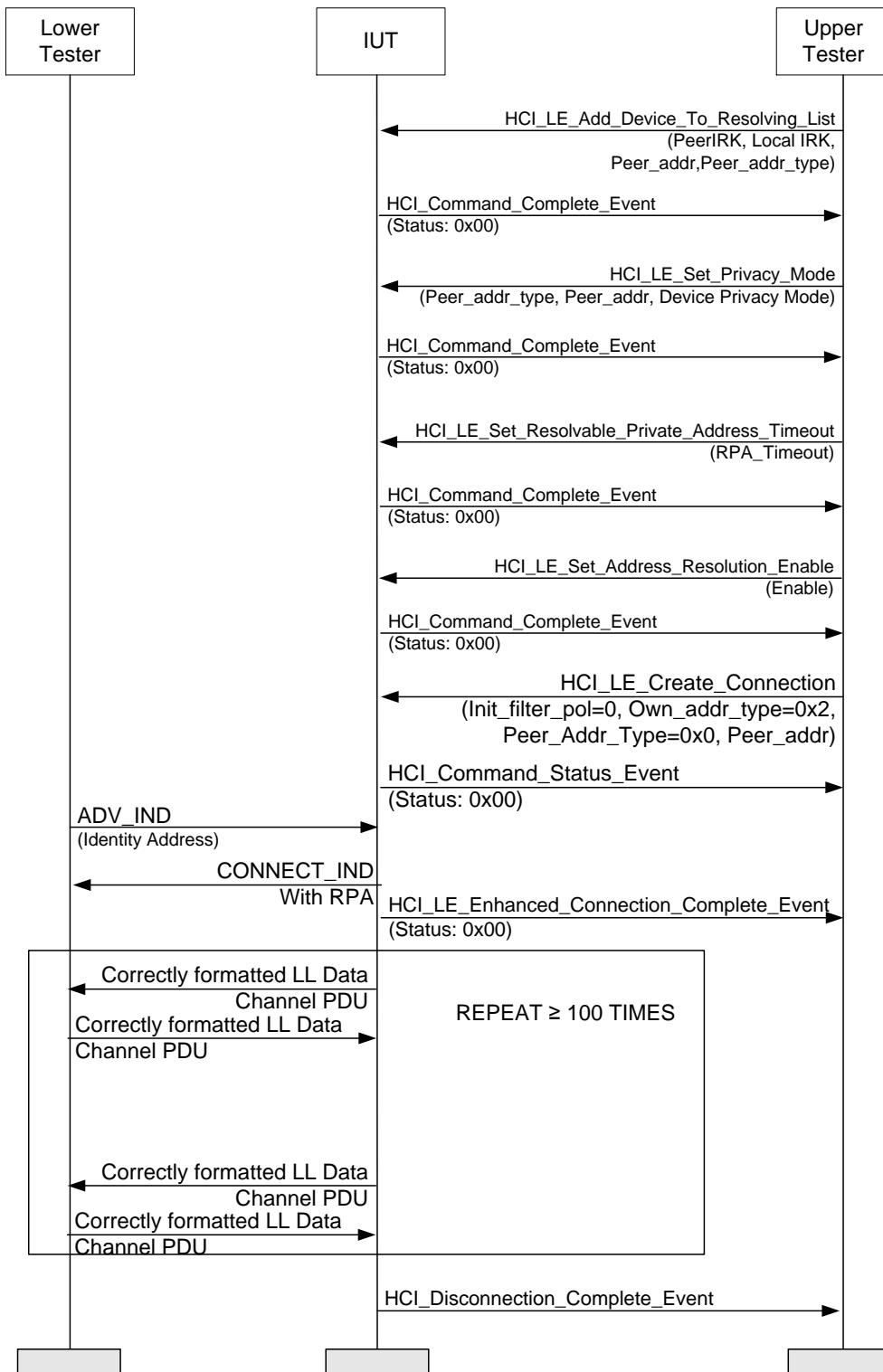


Figure 4.141: LL/CON/INI/BV-20-C [Device Privacy – Connection Establishment using resolving list, Initiator, Accept Identity Address]

1. Configure the Lower Tester to start advertising with its device identity address.
 2. The Upper Tester adds the Lower Tester to the resolving list using an IRK and sets the entry to device privacy mode.

3. The Upper Tester enables the initiator state in the IUT.
4. Lower Tester sends ADV_IND packets, each advertising event, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$, or step 5 occurs.
5. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_IND packets.
6. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester's Identity address and connection interval selected.
7. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
9. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
10. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
11. The Upper Tester commands the IUT (master) to terminate the connection.

- Expected Outcome

Pass Verdict

The IUT receives and creates a connection for connectable advertising with a device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode.

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The IUT reports the connection setup with an HCI event.

4.3.3.20 LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Accept Identity Address]

- Test Purpose

Verify that the IUT when initiating connection establishment connects to a device advertising using its device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode. The Lower Tester uses directed advertising. The IUT may use a public or static random device address.

- Reference

[3] 1.3, 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)



- Test Procedure

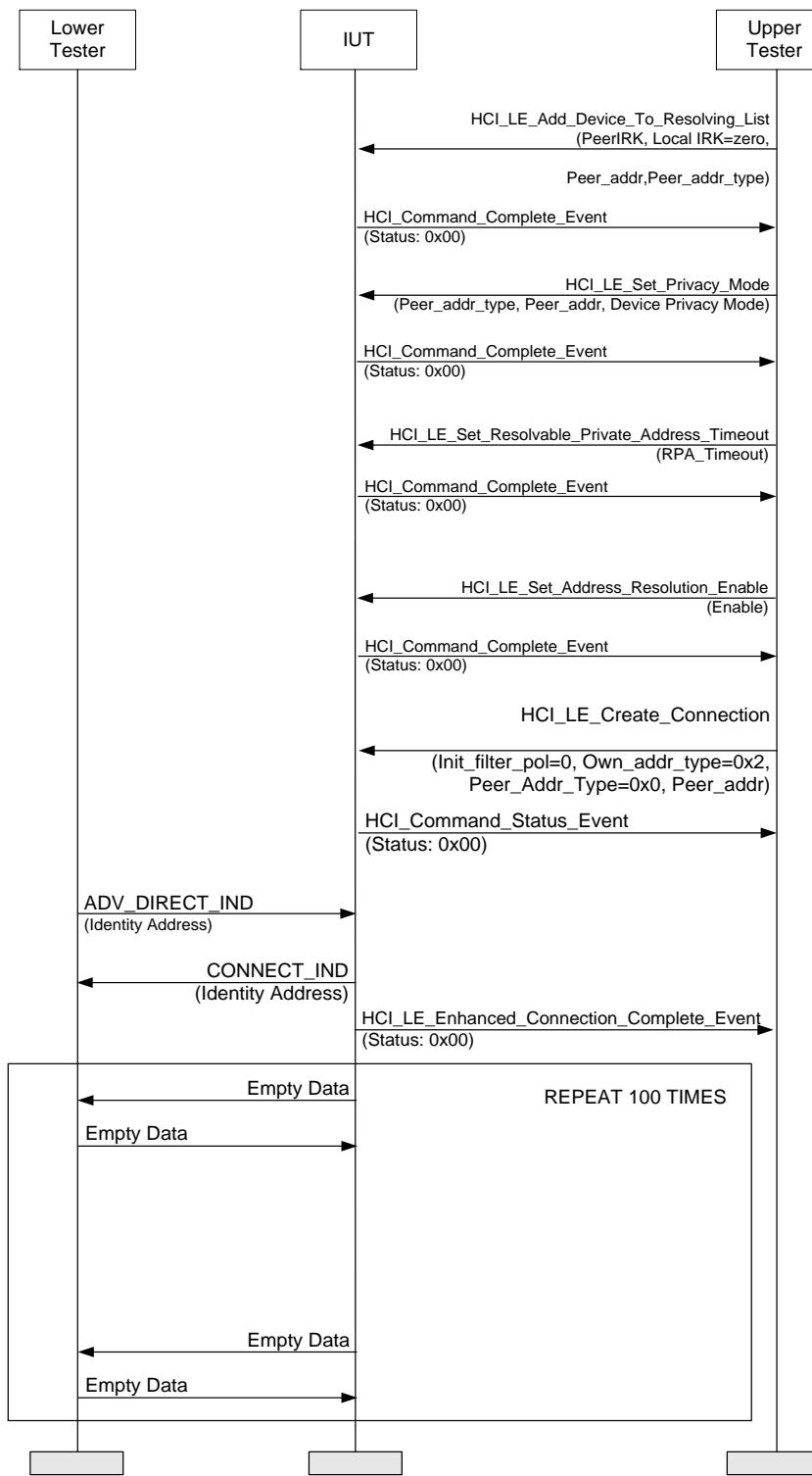


Figure 4.142: LL/CON/INI/BV-21-C [Device Privacy – Connection Establishment using directed advertising and resolving list, Initiator, Accept Identity Address]

- Configure the Lower Tester to start directed advertising with its device identity address.
- The Upper Tester adds the Lower Tester to the resolving list using an IRK and sets the entry to device privacy mode.



3. The Upper Tester enables the initiator state in the IUT.
 4. Lower Tester sends ADV_DIRECT_IND packets directed to the IUT, each advertising event on the selected advertising channel, using the selected advertising interval.
 5. The Lower Tester expects a CONNECT_IND packet TIFS after any of the ADV_DIRECT_IND packets. The AdvA field is the same as the one received in the ADV_DIRECT_IND.
 6. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 7. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 8. The Lower Tester sends a correctly formatted LL Data Channel PDU to the IUT on the same data channel using the acknowledgement scheme.
 9. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 10. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
 11. The Upper Tester commands the IUT (master) to terminate the connection.
- Expected Outcome

Pass Verdict

The IUT receives and creates a connection for directed advertising with a device identity address when the identity address and an associated IRK are in the resolving list using device privacy mode.

The test procedure completes with the IUT sending correctly formatted LL Data Channel PDUs and maintaining the connection.

4.3.3.21 LL/CON/INI/BI-01-C [Connection Initiation Invalid CRC]

- Test Purpose

Tests that an initiator IUT ignores advertising packets with an invalid checksum.

The Lower Tester acts in the advertising state transmitting packets with invalid checksums and observes whether the IUT makes a connection request, then stops the attempt to create a connection.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map,

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, any connection interval, any connection interval, any slave latency, any timeout).

- Test Procedure

Execute the test procedure advertising using a selected scan interval and window once using a common public device address.



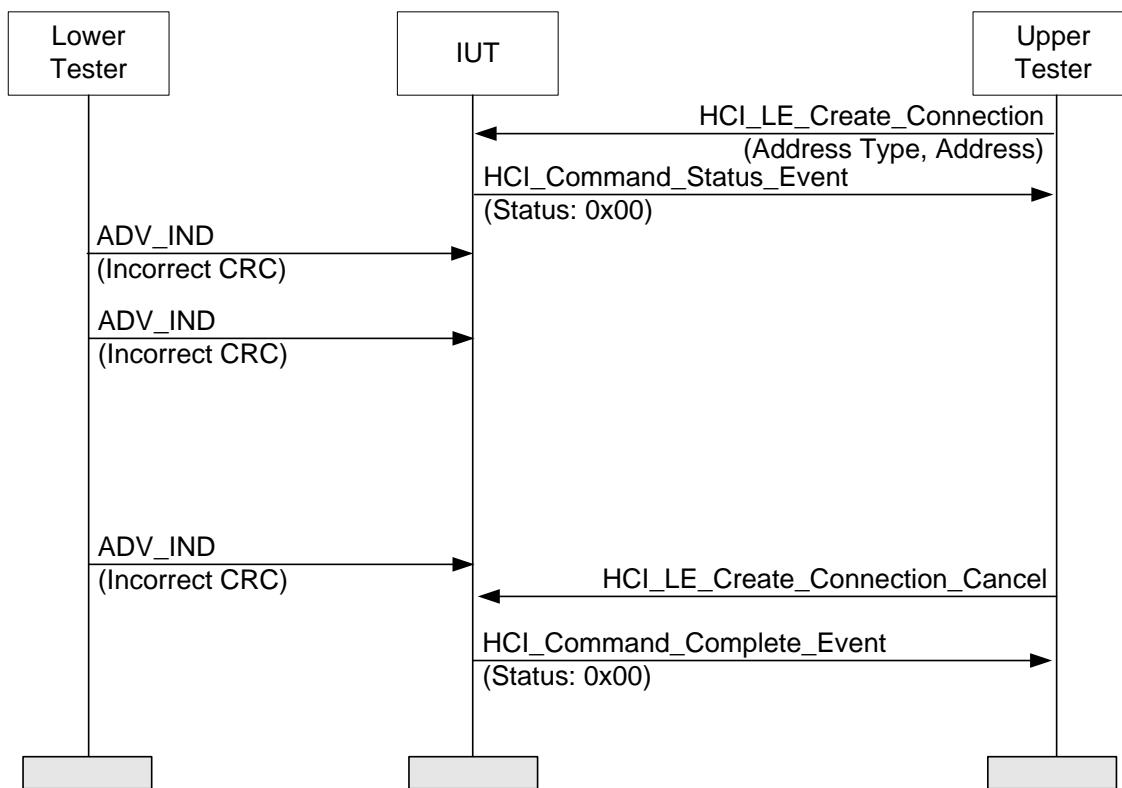


Figure 4.143: LL/CON/INI/BI-01-C [Connection Initiation Invalid CRC]

- Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
 - Upper Tester enables initiator state in the IUT.
 - Lower Tester sends an ADV_IND packet with an invalid checksum, each advertising event using the selected advertising channel only, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$.
 - Lower Tester expects no CONNECT_IND packet TIFS after any of the ADV_IND packets.
 - Upper Tester sends an HCI_LE_Create_Connection_Cancel command to the IUT to stop the connection setup and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The test procedure completes without the IUT sending a connection request and stopping the connection setup procedure reporting it with the HCI event.

4.3.3.22 LL/CON/INI/BI-02-C [Slave Packets Invalid CRC]

- Test Purpose

Tests that an initiator IUT sends a connection request to an advertiser and receiving reply transmissions with invalid checksums from the slave up to the point of expiring the connection supervision timer, considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, and then begins to maintain the connection in the slave role but with packets with an invalid checksum.

- Reference

[3] 4.5.4, 4.51



- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, 0 slave latency, 100 ms timeout).

- Test Procedure

Execute the test procedure using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval to maintain the connection. Use slave latency= 0 and timeout parameter of 100 ms.



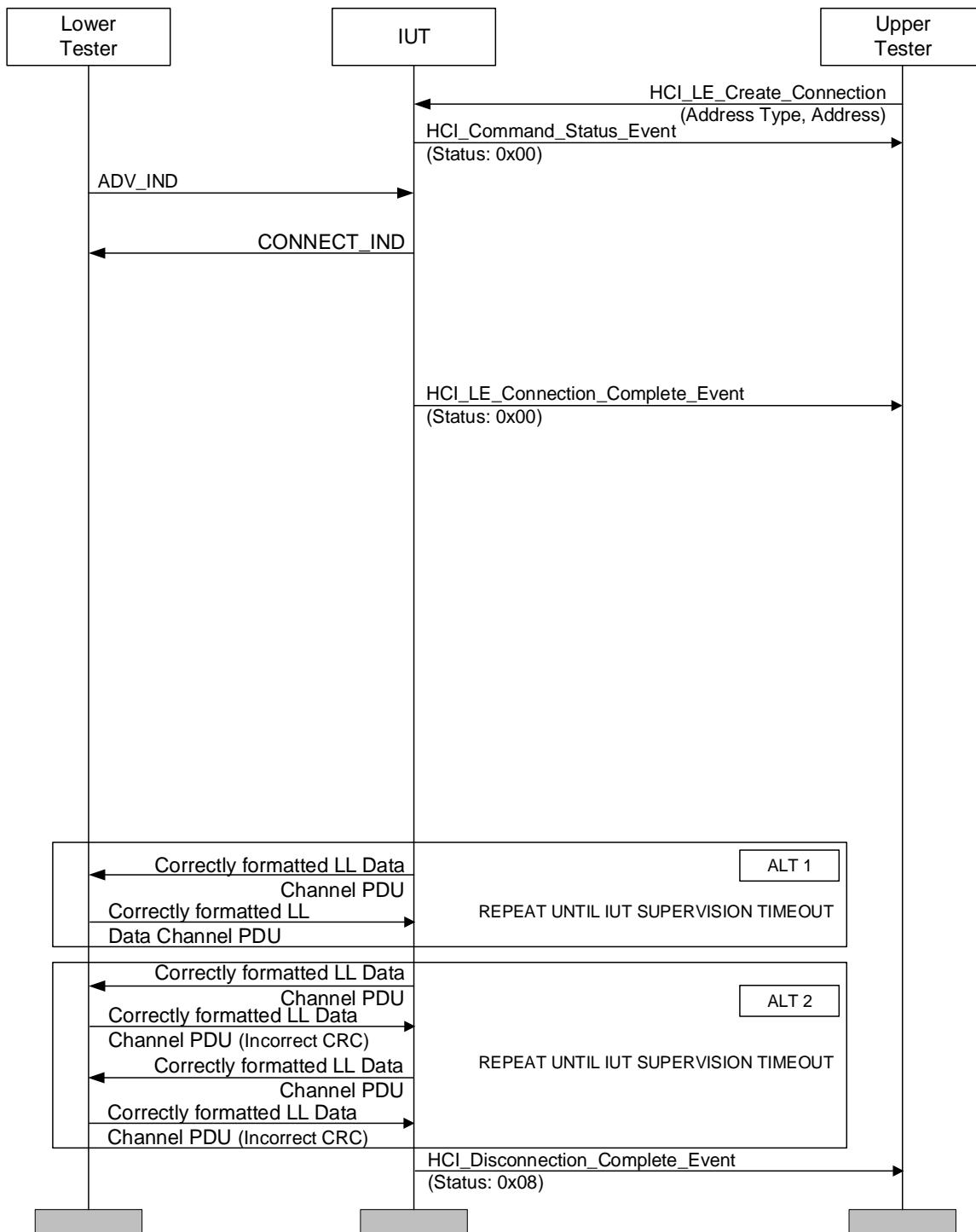


Figure 4.144: LL/CON/INI/BI-02-C [Slave packets Invalid CRC]

- Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
- Upper Tester enables initiator state in the IUT.
- Lower Tester sends an ADV_IND packet with the selected address and 1 byte as data, each advertising event on the selected advertising channel only, using the selected advertising interval. Lower Tester repeats until the time exceeds $4 * \text{scanInterval} + 3 * \text{scanWindow}$ or 4 executions. Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_IND packets.

4. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 5. After the CONNECT_IND packet has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel in the range of transmitWindowOffset and transmitWindowSize.
 6. Lower Tester
 - a) Sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
Receives a correctly formatted LL Data Channel PDU on the same data channel inside the same connection event. If this second Data Channel PDU in the connection event is received, Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
 - b) Expects correctly formatted LL Data Channel PDU on subsequent data channel in the next connection interval of minus (latest plus) maximum clock drift according to the drift rate indicated in the connection request, calculated for the connection interval used.
 - c) Sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel, using the acknowledgement scheme.
 - d) Receives a correctly formatted LL Data Channel PDU on the same connection event. If this second Data Channel PDU in the connect event is received, Lower Tester sends a correctly formatted LL Data Channel PDU with an invalid checksum to the IUT on the same data channel using the acknowledgement scheme.
 7. Repeat Steps 6b to 6c until IUT stops because of supervision timeout.
 8. OR
 9. Repeat Steps 6b to 6d until IUT stops because of supervision timeout.
 10. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating 'CONNECTION TIMEOUT' and with the connection handle matching Step 6.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request, then continuing master transmissions until Connection Supervision expires.

The IUT reports the connection timeout with an HCI event

4.3.3.23 LL/CON/INI/BV-22-C [Connection Initiation, IUT Channel Selection Algorithm #1, Lower Tester Channel Selection Algorithm #2]

- Test Purpose

Tests that an initiator IUT that only supports Channel Selection Algorithm #1 sends a connection request to an advertiser and starts to maintain a connection in the master role when the advertisement indicates support of Channel Selection Algorithm #2. The Lower Tester first acts in the advertising state with ChSel set to one (1), then accepts the connection and starts to maintain it in the slave role, observing the packet and timing from the IUT. The IUT confirms the Channel Selection Algorithm #1 is used for the connection.
- Reference

[10] 4.3.4, 4.4.4, 4.5.3, 4.5.4



- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list is not used, public peer address, lower tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected scan interval and window. Use the common data channel selection parameters and the common connection interval, latency and timeout to maintain the connection.

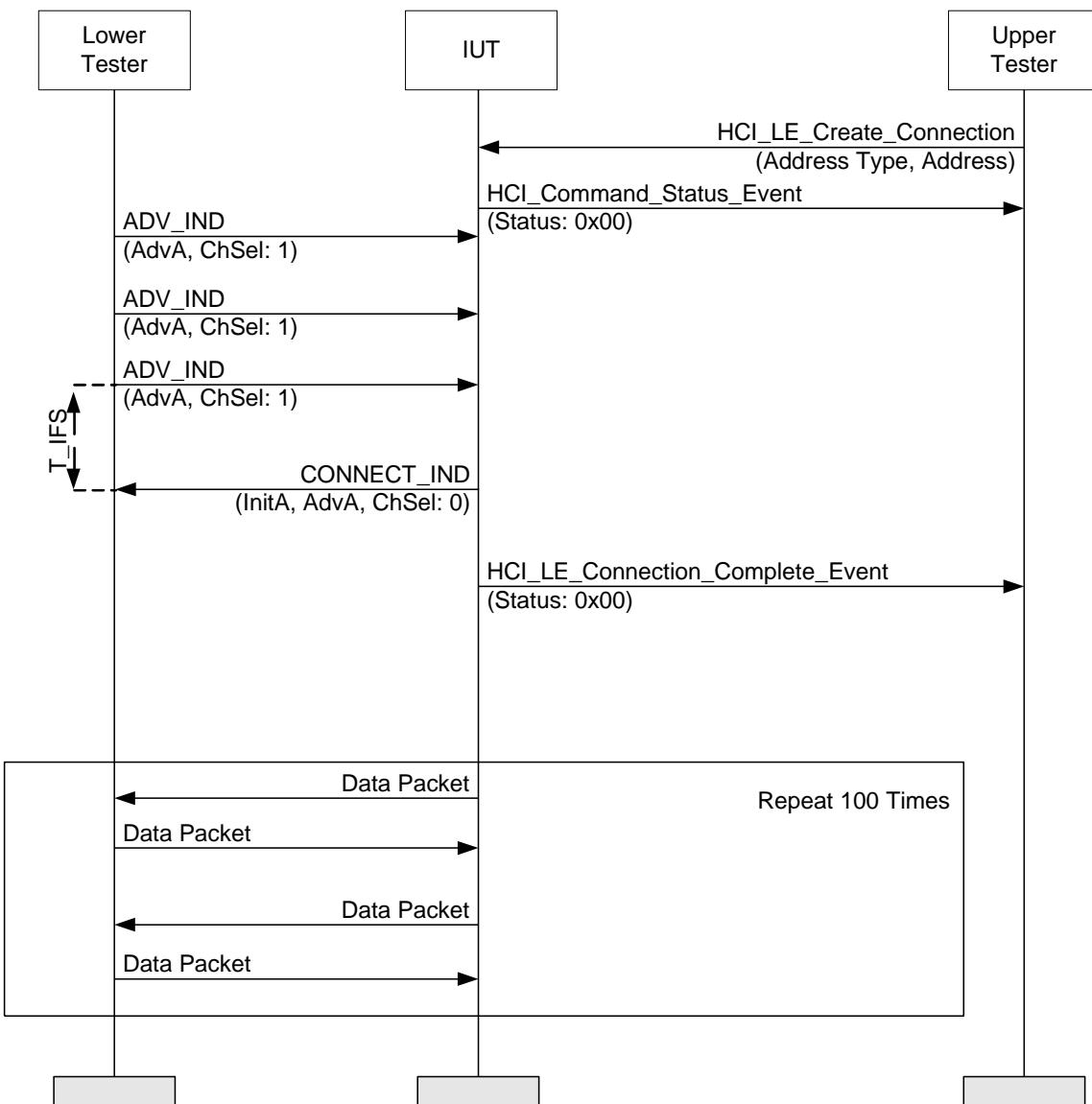


Figure 4.145: LL/CON/INI/BV-22-C [Connection Initiation, Channel Selection Algorithm #1]



1. The Upper Tester sends an HCI_LE_Create_Connection command, with the peer address and address type equal to the ones used by the Lower Tester and the Initiator_Filter_Policy set to 0x00 (use Peer_Address_Type and Peer_Address). The Upper Tester expects an HCI_Command_Status event in response.
 2. The Lower Tester begins advertising using ADV_IND PDUs with ChSel set to 1.
 3. The Lower Tester expects a CONNECT_IND PDU with ChSel set to 0 TIFS after sending any of the ADV_IND PDUs.
 4. The Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the lower tester address and the connection interval selected.
 5. The Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel using the index selected by the LE Channel Selection Algorithm #1 in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize from CONNECT_IND.
 6. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
 7. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 8. Repeat a number of events (100 events) to conclude the timing accuracy.
 9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from Step 4).
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending a connection request and maintaining the connection.

The first event starts within maximum deviation of the allowed transmitWindowOffset and transmitWindowSize, in [Table 4.28](#) above.

The number of timing measurements for event starts from the IUT is at least 100.

The timing deviations detected for packets in active mode are within the 2 μ s range around TIFS.

The connection events' time intervals are within the range expressed for the sleep clock accuracy value.

The difference between the sum of the measured connection events' time intervals and the sum calculated without any drift is equal to or below the limit expressed for the sleep clock accuracy value.

The IUT reports the connection setup with the HCI event.

The access address used by the IUT meets the requirements for access addresses

The IUT sends and receives data using data channel indices selected by the Channel Selection Algorithm #1.

- Notes

Accuracy required for connection events is 0.01 ms. Jitter may contribute to the measurements on a low repetition count. The measurement accuracy is at least 0.001 ms. Drift for the common connection interval of 30 ms used in the test varies on the SCA applied by the IUT from 0.0006 ms to 0.015 ms resulting in the following ranges accepted.



4.3.3.24 LL/CON/INI/BV-23-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled]

- Test Purpose

Verify that the IUT when initiating connection establishment using Whitelist connects to a device advertising using its device identity address when the peer address is in the resolving list with valid local IRK and address resolution is disabled.

- Reference

[3] 4.5.4

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map

State: Initiating (selected scan interval, selected scan window, White List is used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using its Identity Address in the AdvA field of the advertisement packets.

The Peer IRK for IUT is set to zeros.

- Test Procedure

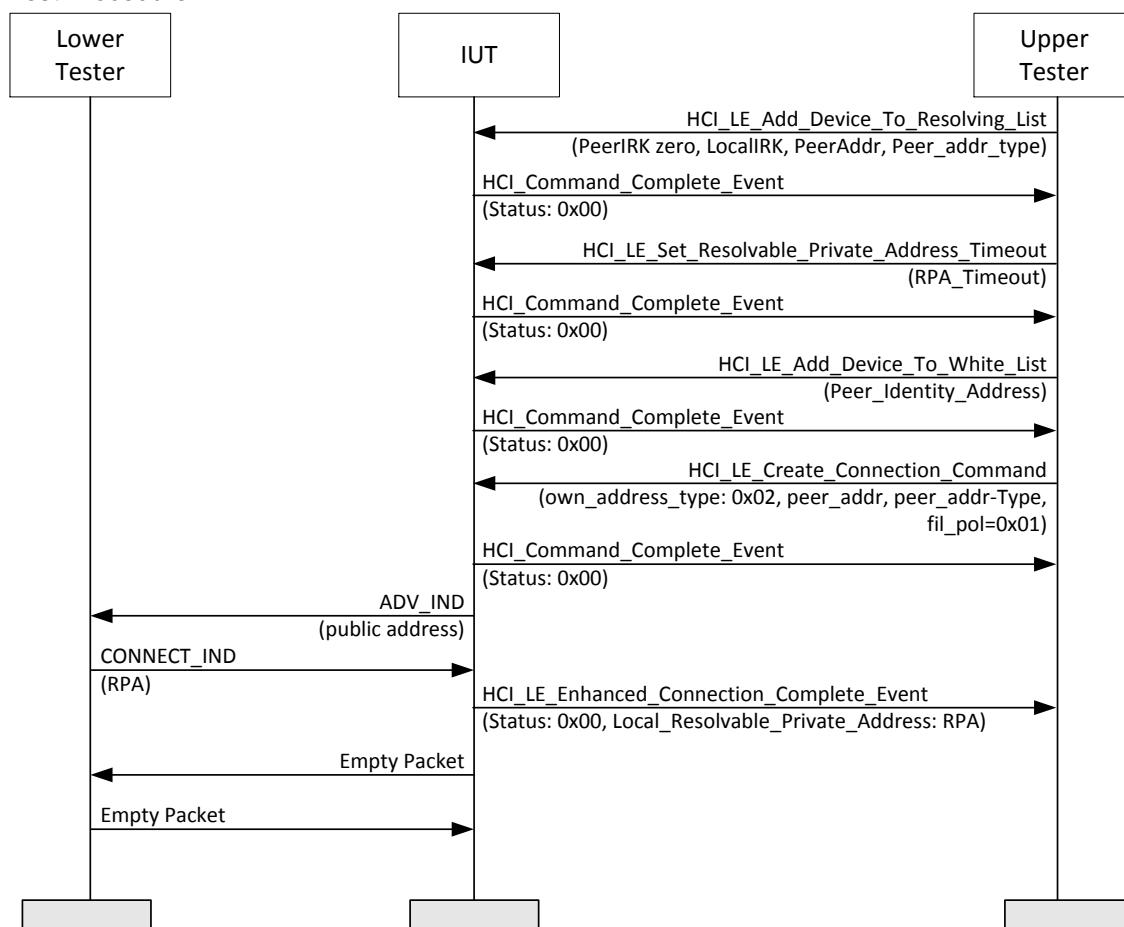


Figure 4.146: LL/CON/INI/BV-23-C [Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled]

1. Configure the Lower Tester to start advertising with its device identity address.
 2. The Upper Tester populates the resolving list with the Lower Tester's address along with 2 more resolving list entries not equal to lower tester with different local IRK for each entry.
 3. The Upper Tester adds the Lower Tester's address to the white list along with 2 other whitelist entries.
 4. The upper tester issues HCI LE Create Connection command to the IUT with own address type set to 0x02, Initiator Filter policy set to 0x01 and peer address set to the lower tester address.
 5. Lower Tester sends adv packets, each advertising event on the selected advertising channel, using the selected advertising interval.
 6. IUT receives the adv packet, and sends connect request with own address set to RPA, generated from the resolving list populated for the peer address.
 7. The Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_IND packets. The AdvA field is the same as the one received in the ADV_IND and INITA as RPA.
 8. Lower Tester resolves private address received from the IUT using assigned IRK.
 9. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 10. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 11. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 12. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

- IUT sends a Connection Indication PDU to Lower Tester with Own Address set to RPA upon receiving an ADV_IND PDU from the Lower Tester advertising with Identity Address when IUT contains a matching entry for the Lower Tester Address in the Resolving List and Address Resolution is disabled at IUT.
- The test procedure completes with the IUT sending a connection request with RPA and maintaining the connection.
- The IUT reports the connection setup with an HCI event.

4.3.3.25 LL/CON/INI/BV-24-C [Network Privacy - Connection Establishment using resolving list with address resolution disabled]

- Test Purpose

Verify that the IUT when initiating connection establishment using resolving list connects to a device advertising using its device identity address when the peer address is in the resolving list with valid local IRK and address resolution is disabled.

- Reference
[3] 4.5.4
- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX,
LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map,
LL_initiator_Channel_Map



State: Initiating (selected scan interval, selected scan window, White List is not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

The Lower Tester is using its Identity Address in the AdVA field of the advertisement packets.

- Test Procedure

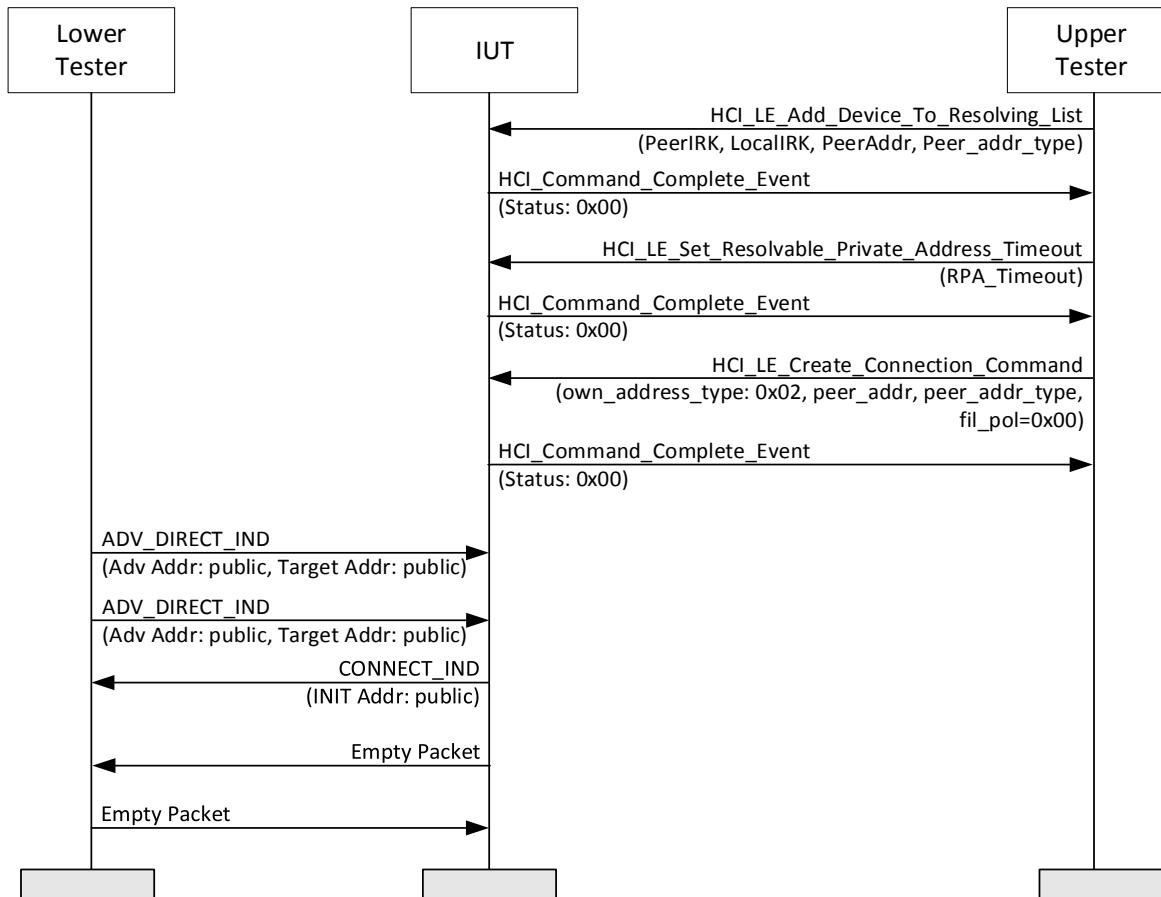


Figure 4.147: LL/CON/INI/BV-24-C [Network Privacy - Connection Establishment using resolving list with address resolution disabled]

1. Configure the Lower Tester to start directed advertising with its device identity address.
 2. The Upper Tester adds the Lower Tester to the resolving list using a valid local and peer IRK.
 3. The upper tester enables issues HCI_LE_Create_Connection command to the IUT with own address type set to 0x02 and peer address set to the lower tester address.
 4. Lower Tester sends directed adv packets, each advertising event on the selected advertising channel, using the selected advertising interval.
 5. IUT receives the adv packet, and sends a connect request with own address set to Identity address.
 6. The Lower Tester expects a CONNECT_IND packet T_IFS after any of the ADV_DIRECT_IND packet. The AdvA field is the same as the one received in the ADV_DIRECT_IND.
 7. Upper Tester expects an HCI_LE_Enhanced_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
 8. After the CONNECT_IND has been received, the Lower Tester expects the first correctly formatted LL Data Channel PDU on the data channel.
 9. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.

10. Repeat a number of events (at least 100 events) to verify that the connection is maintained.
 11. The Upper Tester terminates the connection.
- Expected Outcome
Pass Verdict
 - The IUT receives and creates a connection with own address set to identity address for connectable advertising when the identity address and an associated IRK are in the resolving list with address resolution disabled.
 - The test procedure completes with the IUT sending a connection request with identity address and maintaining the connection.
 - The IUT reports the connection setup with an HCI event.

4.3.4 SLA

Tests that the IUT behaves according to the connection procedures as a slave.

4.3.4.1 LL/CON/SLA/BV-02-C [Slave Asymmetric Connections]

- Test Purpose
Tests that a slave IUT responds in all events in a connection where the slave latency parameter is not zero.
The Lower Tester acts in the master role in the connection and observes the slave packet timing and packet contents on the data channels in use.
- Reference
[3] 4.5
- Initial Condition
Parameters: LL_slave_connSlaveLatency_MIN
State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)
- Test Procedure
Execute the test procedure using the common connection parameters, with slave latency of 5 and a timeout value of 200 ms.



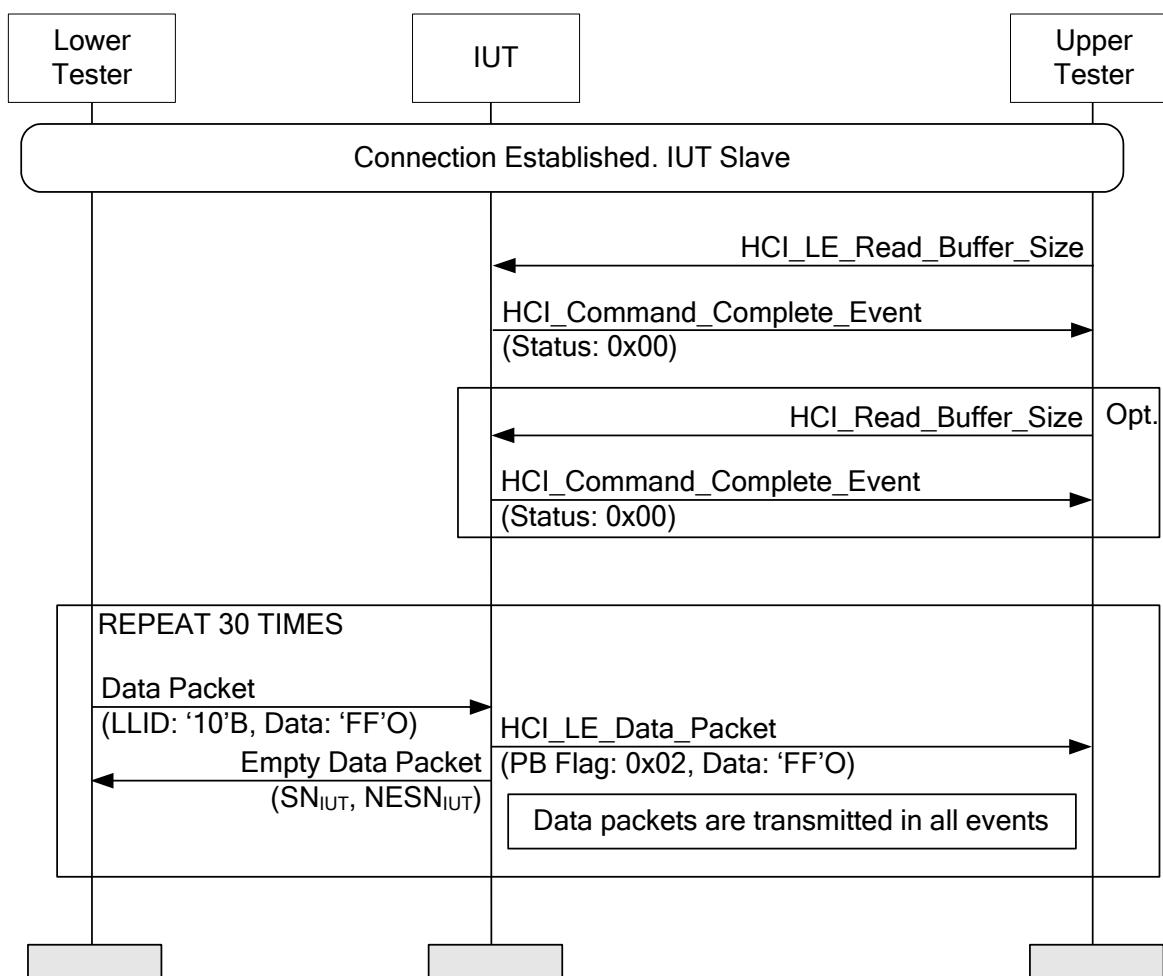


Figure 4.148: LL/CON/SLA/BV-02-C [Slave asymmetric connections]

1. Upper Tester sends **HCI_LE_Read_Buffer_Size** and expects an **HCI_Command_Complete_Event**.
 2. Optional: Upper Tester sends **HCI_Read_Buffer_Size** and expects an **HCI_Command_Complete_Event**.
 3. Lower Tester sends a DATA packet once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received matches the next SN and by using the next NESN where the SN in the previous packet correctly received matches the current NESN.
 4. Upper Tester expects an **HCI_Le_Data_Packet** from the IUT containing a data element sent in 3 and with the **Packet_Boundary_Flag** flag set.
 5. Lower Tester expects an empty data packet T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Allow mismatches for packets not received correctly.
 6. Repeat steps 3-5 30 times.
 7. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- **Test Condition**

The parameters in this test are calculated for a BER of 0.1 % or better.

- Expected Outcome

Pass Verdict

The IUT responds to Lower Tester according to latency value (at least 5 packets in 30).

- Notes

The error rate calculation considers only the access address resulting in that roughly 6 % of symmetric events can be missed by a slave device.

4.3.4.2 LL/CON/SLA/BV-04-C [Slave Sending Data]

- Test Purpose

Tests that a slave IUT is able to send data to a master device.

The Lower Tester acts in the master role in the connection, submits data from host of the IUT to transmit and using the acknowledgement scheme, observes the data in the packets from the IUT.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_slave_payload_length_MIN, LL_slave_payload_length_MAX

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



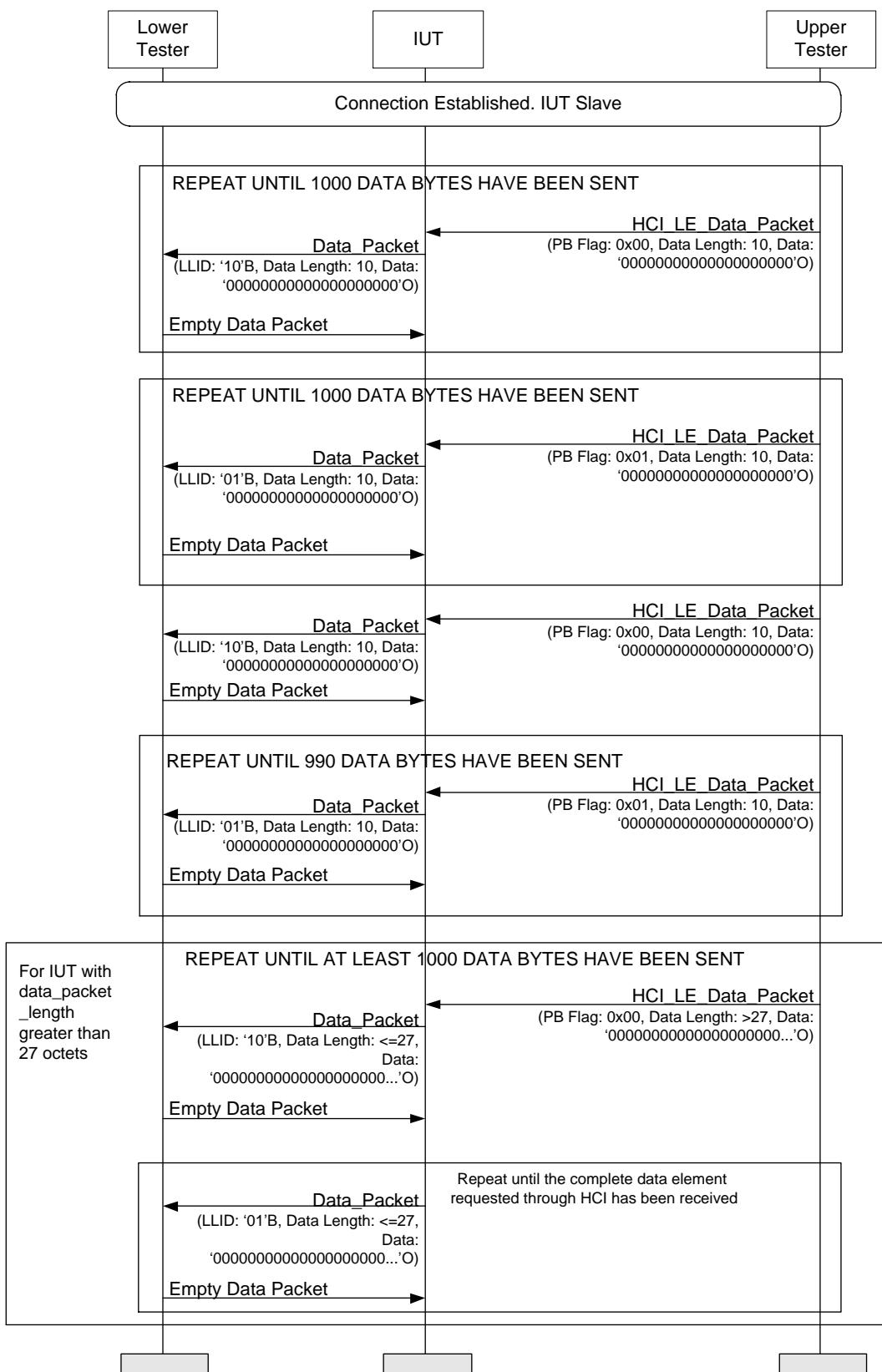


Figure 4.149: LL/CON/SLA/BV-04-C [Slave sending data]



1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted. It expects the appropriate HCI_Number.Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
2. Lower Tester expects a DATA packet from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1.
3. Lower Tester sends an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
4. Repeat steps 2 and 3 until all elements submitted in step 1 have been received.
5. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x01 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) is successfully submitted. It expects the appropriate HCI_Number.Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
6. Lower Tester expects a DATA packet from the IUT, with the LLID field set to 0x01 and containing a data element submitted in step 5. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
7. Repeat step 6 until all elements submitted in step 5 have been received.
8. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, with Packet_Boundary_Flag flag equal to 0x00 in the first packet and equal to 0x01 in the following packets until the selected number of octets (1000) are successfully submitted. It expects the appropriate HCI_Number.Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
9. Lower Tester expects a DATA packet from the IUT, containing a data element submitted in step 8: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following packets. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
10. Repeat step 9 until all data sent in step 8 have been reported.
11. If the devices has a data_packet_length (as defined in [Figure 4.4: Buffer Size Read Preamble Steps](#)) less than or equal to 27 octets, skip to step 14; otherwise continue with step 12.
12. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 and data elements with the value 0x00, using a data total length generated randomly within the interval from 28 to the data_packet_length (as defined in [Figure 4.4: Buffer Size Read Preamble Steps](#)), until the selected number of octets (1000) are successfully submitted.
13. For each HCI Data packet command sent by the Upper Tester in step 12, the Lower Tester expects two or more DATA packet from the IUT, with the LLID field set to 0x02 in the first packet with data and set to 0x01 in the following packets, until all the data elements sent to the IUT in step 12 have been received from the IUT. The Lower Tester sends an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
14. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the preamble step execution).



- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending all of the data.

The IUT maintains correct sequence of the fragmentation flags as specified in Section 4.1.6.

4.3.4.3 LL/CON/SLA/BV-05-C [Slave Receiving Data]

Tests that a slave IUT is able to receive data from a master device.

The Lower Tester acts in the master role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- Reference

[3] 4.5

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value).

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



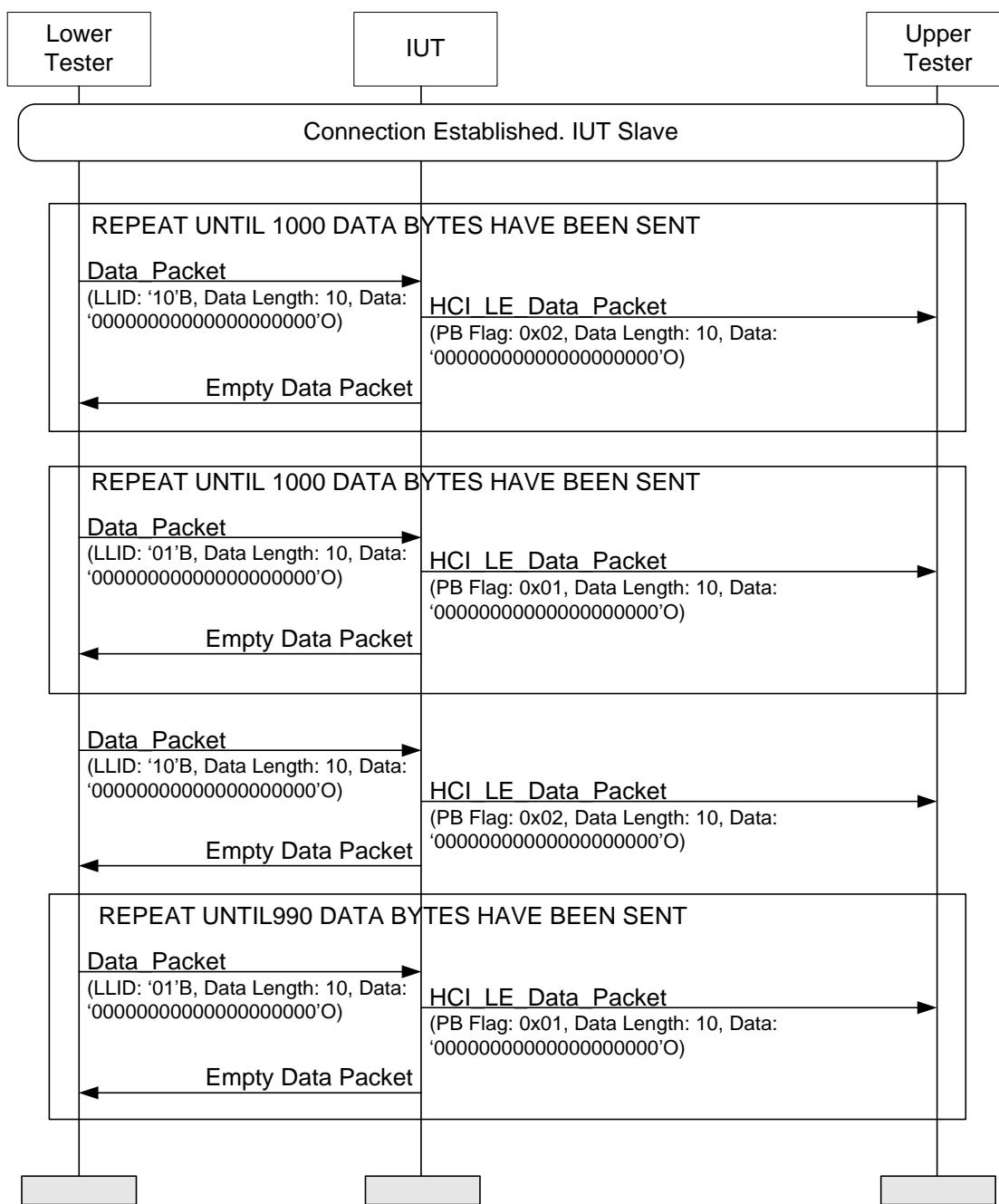


Figure 4.150: LL/CON/SLA/BV-05-C [Slave receiving data]

- Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
- Lower Tester sends a DATA packet once a connection interval to the IUT, using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00. Lower Tester expects a DATA packet in response from the IUT.
- Repeat step 2 until all data sent in step 1 has been reported.
- Upper Tester expects an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 1 with the Packet_Boundary_Flag flag set to 0x02.
- Configure Lower Tester to send 100 data packets with the LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.



6. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x01, using a payload length of 10 with the payload octets set to 0x00. Repeat until all data sent in step 5 have been reported.
7. Upper Tester expects HCI_LE_Data_Packet events from the IUT containing the data element sent in step 6 with the Packet_Boundary_Flag flag set to 0x01.
8. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 in the first packet and 0x01 in the following. Using a payload length of 10 with the payload octets set to 0x00.
9. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02 in the first packet with data and 0x01 in the following, and using a payload length of 10 with the payload octets set to 0x00. Repeat until all data sent in step 8 has been reported.
10. Upper Tester expects an HCI_LE_Data_Packet event from the IUT with Packet_Boundary_Flag flag set to 0x02 and additional HCI_LE_Data_Packets events with Packet_Boundary_Flag flag set to 0x00. Payload octets shall be 0x00.

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent,

The IUT reports all data correctly with HCI_Data_Packet events using the HCI fragmentation flags as specified in Section 4.1.6.

4.3.4.4 LL/CON/SLA/BV-06-C [Slave Sending and Receiving Data]

- Test Purpose

Tests that a slave IUT is able to send and receive data to/from a master device.

The Lower Tester acts in the master role in the connection, both submits data for the IUT to transmit and sends data to it according to the acknowledgement scheme and observes the data received and reported to the host of the IUT.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_slave_payload_length_MIN, LL_slave_payload_length_MAX

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps. The number of octets transferred is over the data packet length of the IUT, with a minimum of 1000 octets.



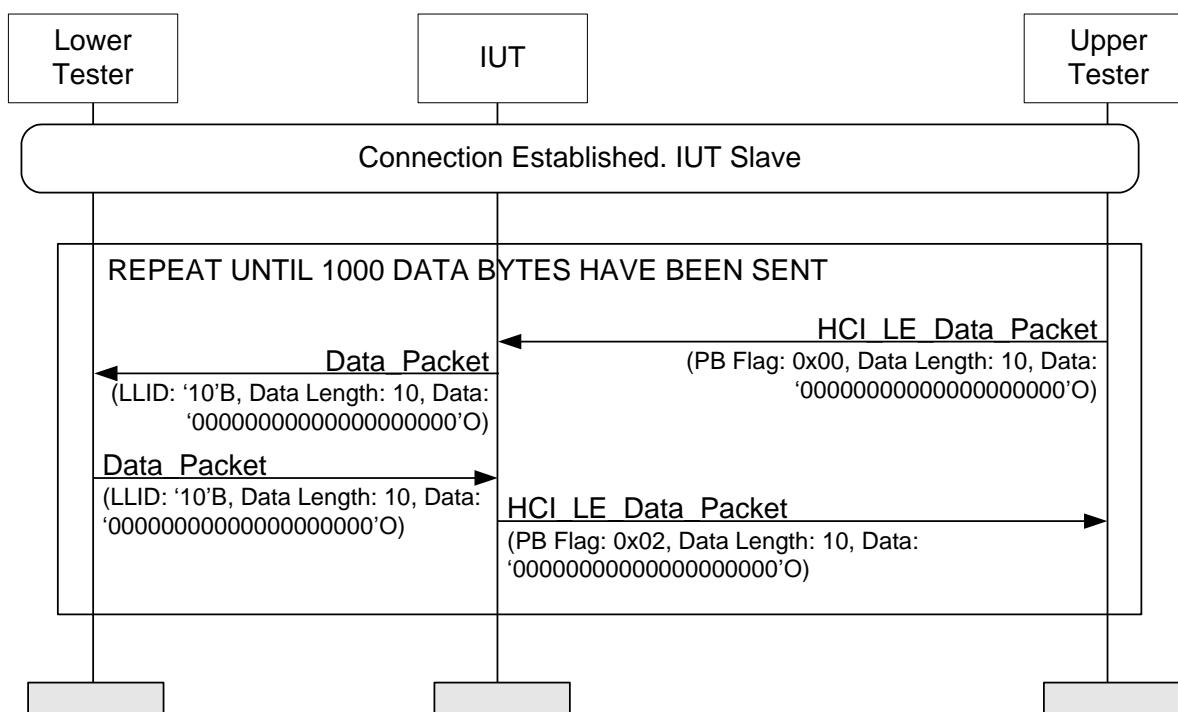


Figure 4.151: LL/CON/SLA/BV-06-C [Slave receiving data and receiving data]

1. Upper Tester submits data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set to 0x00 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
 2. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
 3. Lower Tester sends a DATA packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00.
 4. Lower Tester expects a DATA packet in response from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1.
 5. Repeat steps 3-4 until all data sent in step 1 and 2 have been received/reported
 6. Upper Tester expects an HCI_Data_Packet event from the IUT containing a data element sent in step 1: with the Packet_Boundary_Flag flag set to 0x02 until all data sent in 1 have been reported.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent and reporting all data received,

The IUT sends the data preserving the fragmentation flags in the LLID field, as specified in Section 4.1.6.

The IUT reports the data preserving the upper layer messages in HCI_Data_Packet events.



4.3.4.5 LL/CON/SLA/BV-10-C [Accepting Parameter Update]

- Test Purpose

Tests that a slave IUT accepts a connection parameter update packet from a master device and starts using the new parameters in the event requested.

The Lower Tester acts in the master role in the connection, sending a connection parameter update packet to the IUT until it accepts it, then takes the new parameters into use. The Lower Tester observes the slave responding in all events until the update.

- Reference

[3] 5.1.1

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common data channel selection parameters. Make the event count value minimum when changing from a short interval and maximum when changing from a long interval:

An update from the common connection interval values to the intermediate values using an event count of 100,

An update from intermediate connection values (100 ms interval, 2 slave latency, 3 s connection supervision timeout) back to the common connection interval using an event count of 10.



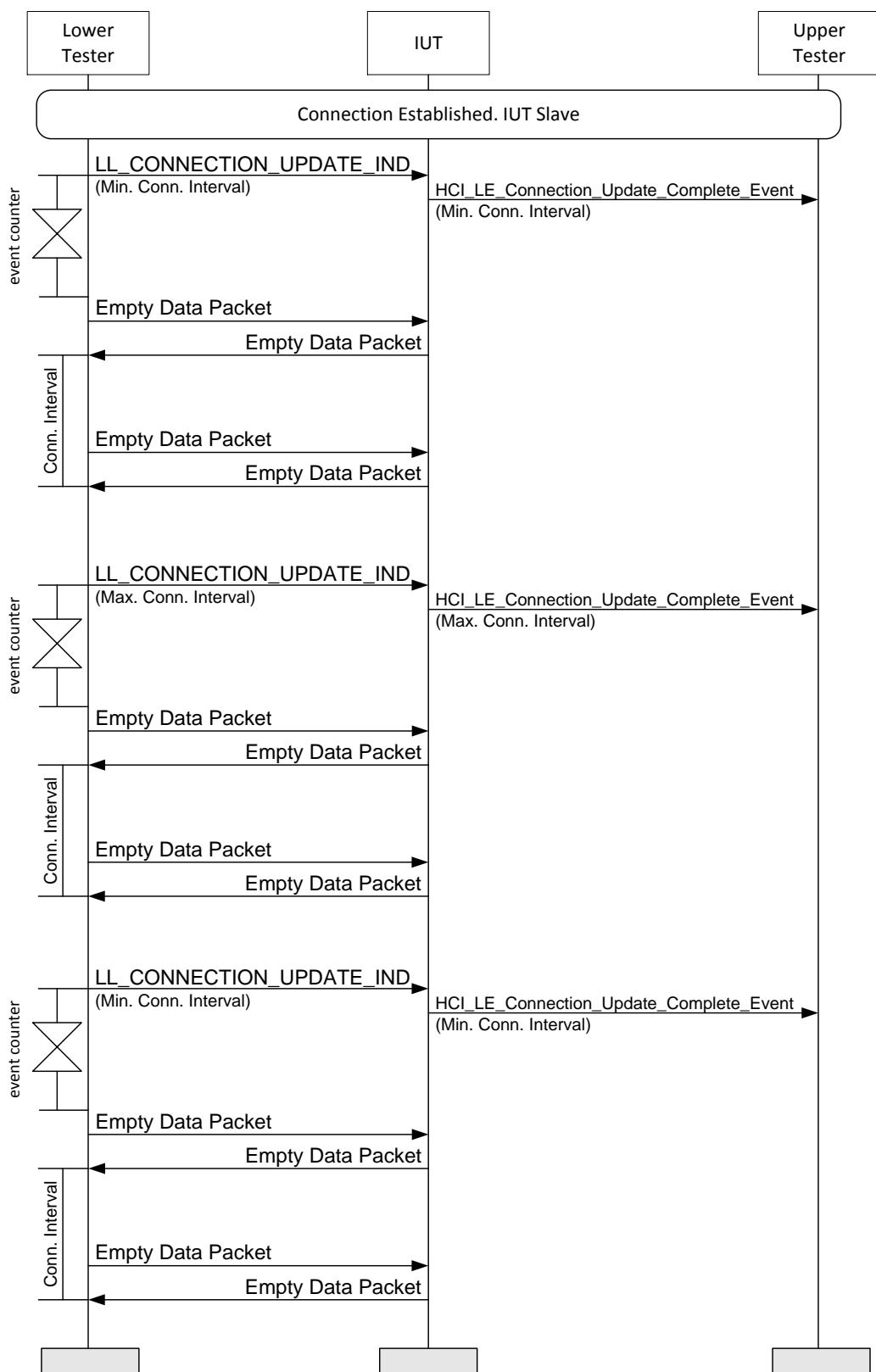


Figure 4.152: LL/CON/SLA/BV-10-C [Accepting parameter update]

1. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the minimum connection interval, no latency, intermediate connection supervision timeout, and



- event count value maximum values. Lower Tester expects a packet from the IUT acknowledging the connection update request.
2. Lower Tester sends empty DATA packets to the IUT, expecting responses until the event count matches the indicated time of connection update:
 3. At the time of the update start maintaining the connection with the new parameters.
 4. Interleave with step 3: Upper Tester expects an HCI_LE_Connection_Update_Complete event from the IUT containing the new connection parameters.
 5. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum connection interval, no latency, intermediate connection supervision timeout, and event count value minimum values. Expect a packet from the IUT acknowledging the connection update request.
 6. Repeat steps 2-4
 7. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the minimum connection interval, no latency, intermediate connection supervision timeout, and event count value maximum values. Expect a packet from the IUT acknowledging the connection update request.
 8. Repeat steps 2-4.
- Test Condition
- The parameters in this test are calculated for a BER of 0.1 % or better.
- Expected Outcome
- Pass Verdict
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event,
- The IUT responds in at least 65 of the 100 events before the assigned event, when observed in the connection update from the short connection interval to the long interval,
- The IUT reports the new connection parameters with an HCI event.
- Notes
- The response rate (the specific percentage 88.7 %) from the IUT is assuming a BER of 0.1 % on the preamble and access address from the Lower Tester and the empty data packet from the IUT, for a length of 120 bits, with a near 100% confidence level in the test procedure gives around 65 out of 100. This accuracy is rough, it may be improved by:
- Increasing the number of events (the packet count) in the update, but this results in over 1 s testing time,
 - Decreasing the reliability of the test case, making fewer repetitions under in conditions of interference.
 - Neither of the above is desirable, therefore as the 80 % is more than the slave latency it is a satisfactory result.

4.3.4.6 LL/CON/SLA/BV-11-C [Slave Sending Termination]

- Test Purpose

Tests that a slave IUT is able to terminate a connection by sending the termination packet.

The Lower Tester acts in the master role in the connection, receiving the termination packet from the IUT. The Lower Tester observes that there are no more packets from the slave once the termination packet is acknowledged.



- Reference
[3] 5.1.6
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, 100 ms timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using 0 slave latency and a 100 ms timeout.

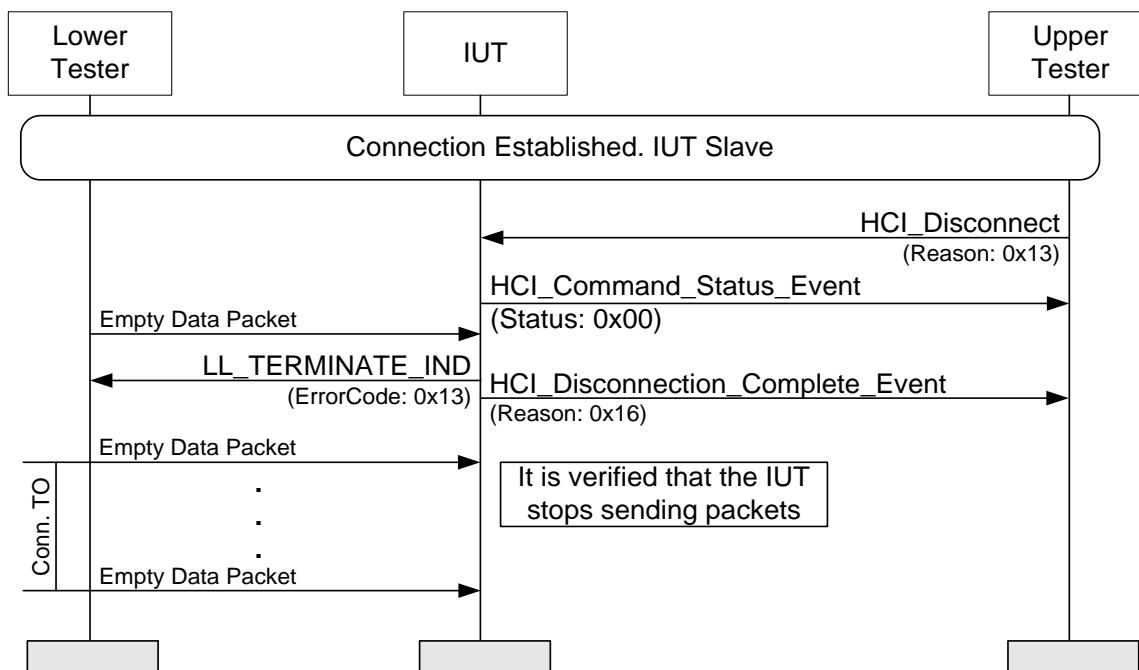


Figure 4.153: LL/CON/SLA/BV-11-C [Slave sending termination]

1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status in response.
 2. Lower Tester expects the IUT to respond to a master transmission with a TERMINATE_IND packet. Lower Tester acknowledges the termination packet with an empty DATA packet in the following event.
 3. Lower Tester sends empty DATA packets up to a time equal to the connection supervision timeout and expects no response to the master transmissions:
 4. Interleave with step 3: Upper Tester expects an HCI_Disconnection_Complete event from the IUT indicating that the connection termination procedure requested in step 1 was successful.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT sending termination packets (success error code) until acknowledgement from the Lower Tester.

The IUT stops maintaining the connection once the TERMINATE_IND packet has been acknowledged by the Lower Tester.

The IUT reports the connection termination with an HCI event.



The IUT stops sending TERMINATE_IND packets when T_Terminate timer expires.

4.3.4.7 LL/CON/SLA/BV-12-C [Slave Accepting Termination]

- Test Purpose

Test that a slave IUT accepts the termination from the master transmissions.

The Lower Tester acts in the master role in the connection, sending the termination packets until receiving acknowledgement from the IUT.

- Reference

[3] 5.1.6

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using 0 slave latency and a 100 ms timeout.

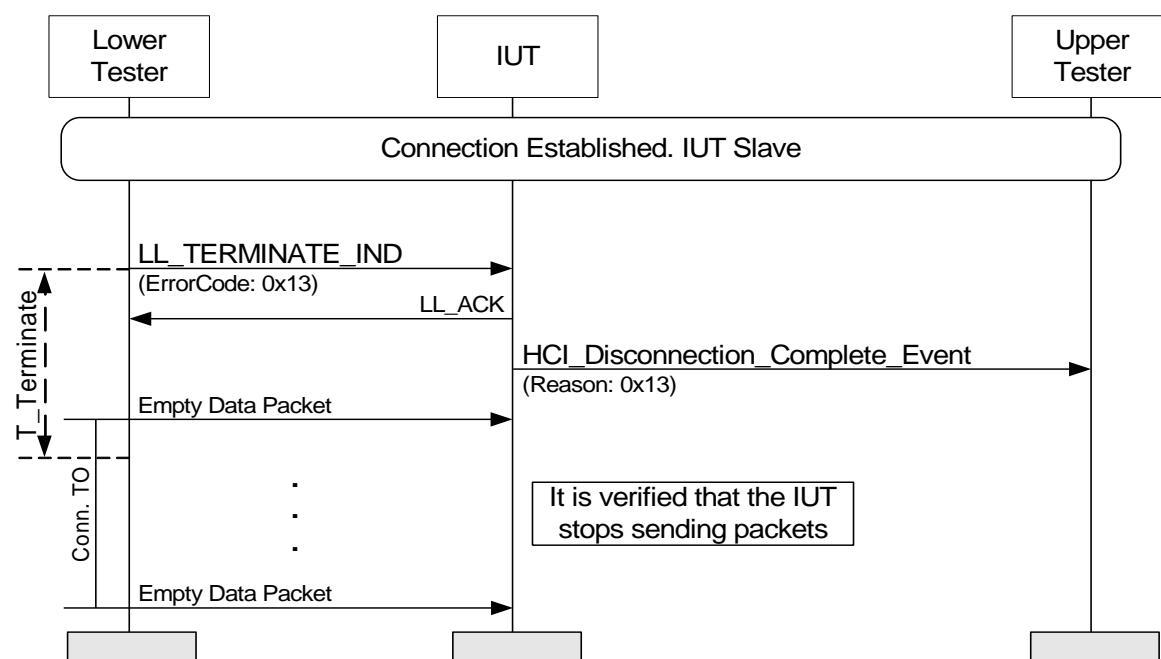


Figure 4.154: LL/CON/SLA/BV-12-C [Slave accepting termination]

1. Lower Tester sends a TERMINATE_IND packet to the IUT and expects an empty DATA packet in response acknowledging the TERMINATE_IND packet.
2. Lower Tester sends empty DATA packets up to a time equal to the connection supervision timeout and expects no response to the master transmissions.
3. Interleave with step 2: Upper Tester expects an HCI_Disconnection_Complete event from the IUT indicating termination requested by the peer device and containing the connection handle from the preamble.



- Expected Outcome

Pass Verdict

The IUT acknowledges one termination packet, before T_Terminate timer expires

The IUT stops maintaining the connection once it has acknowledged the TERMINATE_IND packet.

The IUT reports the connection termination with an HCI event.

4.3.4.8 LL/CON/SLA/BV-13-C [Slave Supervision Timer]

- Test Purpose

Tests that a slave IUT terminates the connection from the connection supervision timer.

The Lower Tester acts in the master role in the connection, stops transmitting events and observes the IUT notifying that the connection has been terminated.

- Reference

[3] 4.5.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure

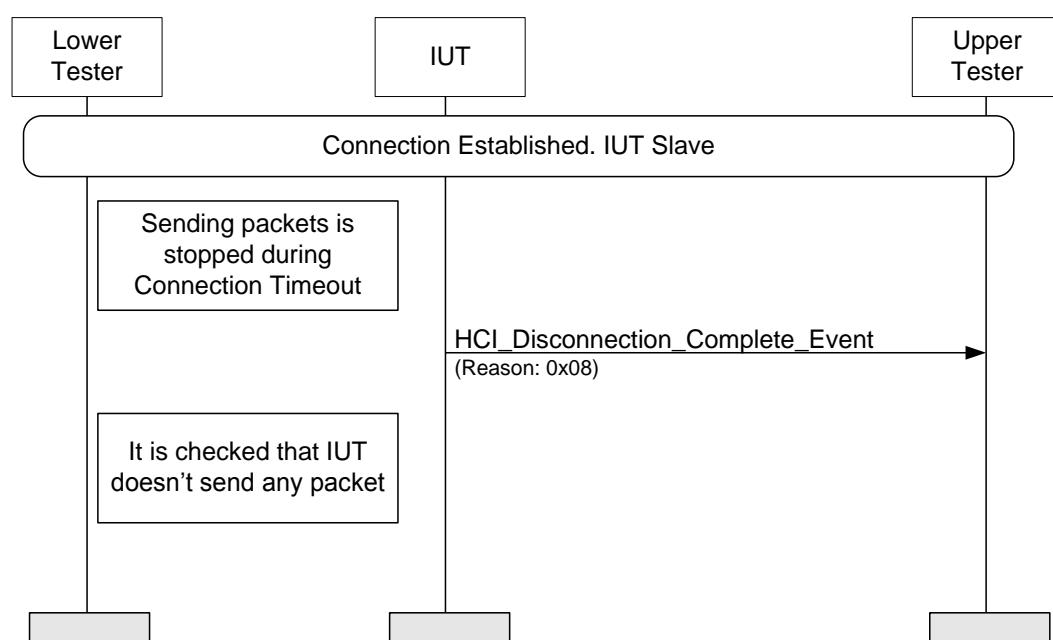


Figure 4.155: LL/CON/SLA/BV-13-C [Slave supervision timer]

1. The connection is established using the maximum connection supervision timer and connection interval values supported.
2. Lower Tester stops maintaining the connection as a master after a specific event.
3. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating a connection supervision timeout and containing the connection handle used, after the last



- connection event before the connection supervision timeout expires (time equal to the connection supervision timeout selected from the last event transmitted).
4. The connection is established using the minimum connection supervision timer and connection interval values supported.
 5. Repeat steps 2-3.
- Expected Outcome

Pass Verdict

The IUT produces the connection termination HCI event after the selected timeout values for the connSupervisionTimeout from the event.

- Notes
- The required accuracy for the connection supervision timer is milliseconds. The drift with 50 ppm to the longest setting of connection supervision timeout, milliseconds is affected. Jitter does not affect the accuracy required.

The deviation by drift is in the order of 1 ms, making that the measurement accuracy required. Measurement results are rounded to tens of milliseconds for the maximum connection supervision timeout value and to milliseconds to the minimum.

4.3.4.9 LL/CON/SLA/BV-14-C [Feature Setup Response]

- Test Purpose
- Tests that a connected slave IUT performs the feature setup procedure, activating the correct features when requested.
- The Lower Tester acts in the master role in a maintained connection, transmits the request to perform feature setup and observes the IUT responding.
- Reference
- [3] 5.1.4
- Initial Condition
- State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, selected slave latency, selected timeout, common channel map, any SCA value).



- Test Procedure

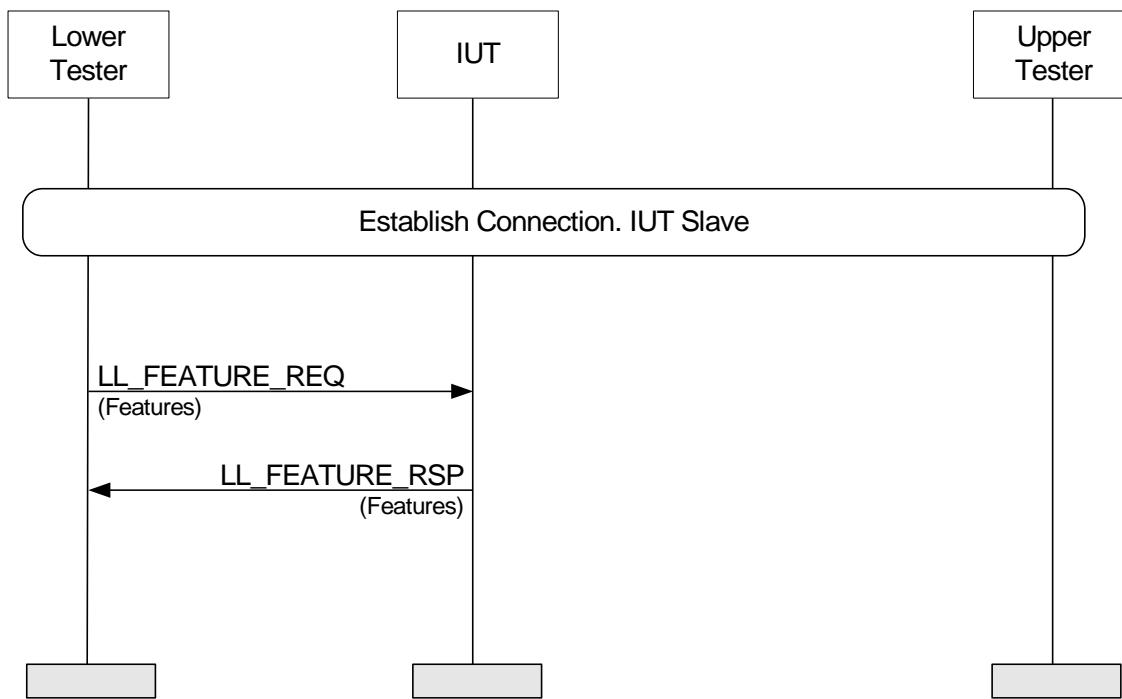


Figure 4.156: LL/CON/SLA/BV-14-C [Feature Setup response]

1. Lower Tester sends an LL_FEATURE_REQ PDU including the Lower Tester's feature set and waits for an LL_FEATURE_RSP PDU.
2. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble step execution).

- Expected Outcome

Pass Verdict

All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT responding with the feature information response.

4.3.4.10 LL/CON/SLA/BV-15-C [Slave Retransmission Request]

- Test Purpose

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme while receiving invalid checksums in data packets.

The Lower Tester acts in the master role, starting events using the default connection parameters, using invalid checksums to prompt a repeated retransmission request from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)



- Test Procedure

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

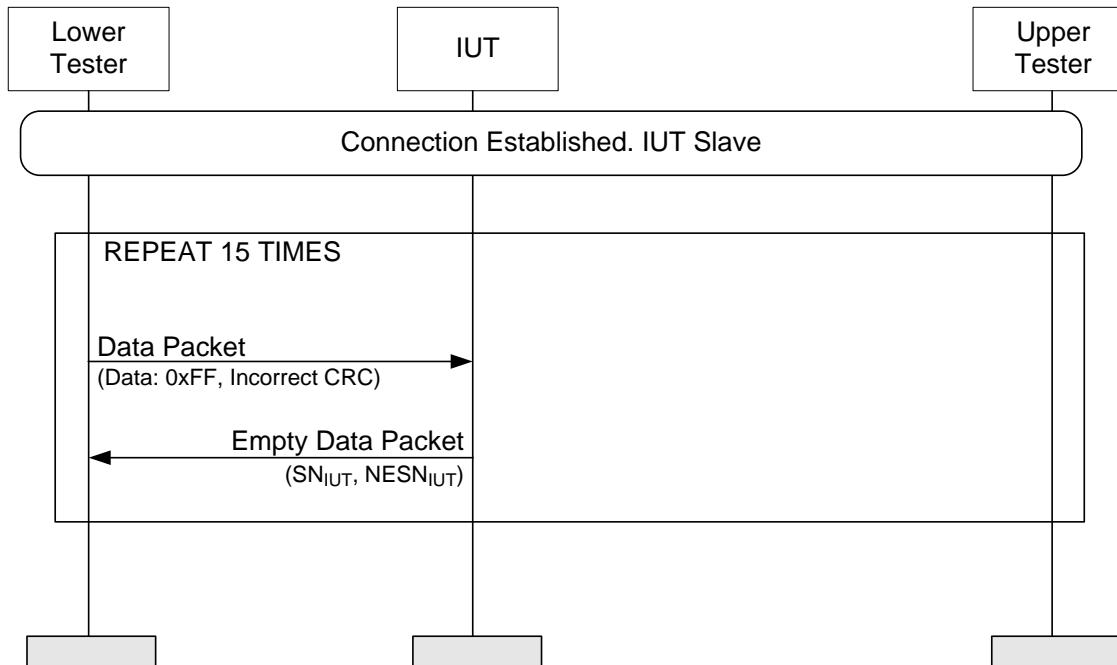


Figure 4.157: LL/CON/SLA/BV-15-C [Slave retransmission request]

1. Lower Tester sends a DATA packet with invalid checksum once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet received is the next SN and by using the next NESN where the current NESN matches the SN in the previous packet correctly received.
2. In events where the IUT is required to listen, Lower Tester expects a DATA packet T_IFS after the packet sent. Lower Tester expects the IUT to indicate in packets received correctly that the previous packet was not received correctly by the SN matching the current NESN and the NESN matching the current SN.
3. Repeat steps 1-2 15 times.
4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).

- Expected Outcome

Pass Verdict

The IUT transmits negative acknowledgements when sent a packet with an invalid CRC in 1.

4.3.4.11 LL/CON/SLA/BV-16-C [Slave Retransmission]

- Test Purpose

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and retransmit a data packet on a negative acknowledgement.



The Lower Tester acts in the master role starting events using the default connection parameters and using negative acknowledgements prompts repeated retransmission requests from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

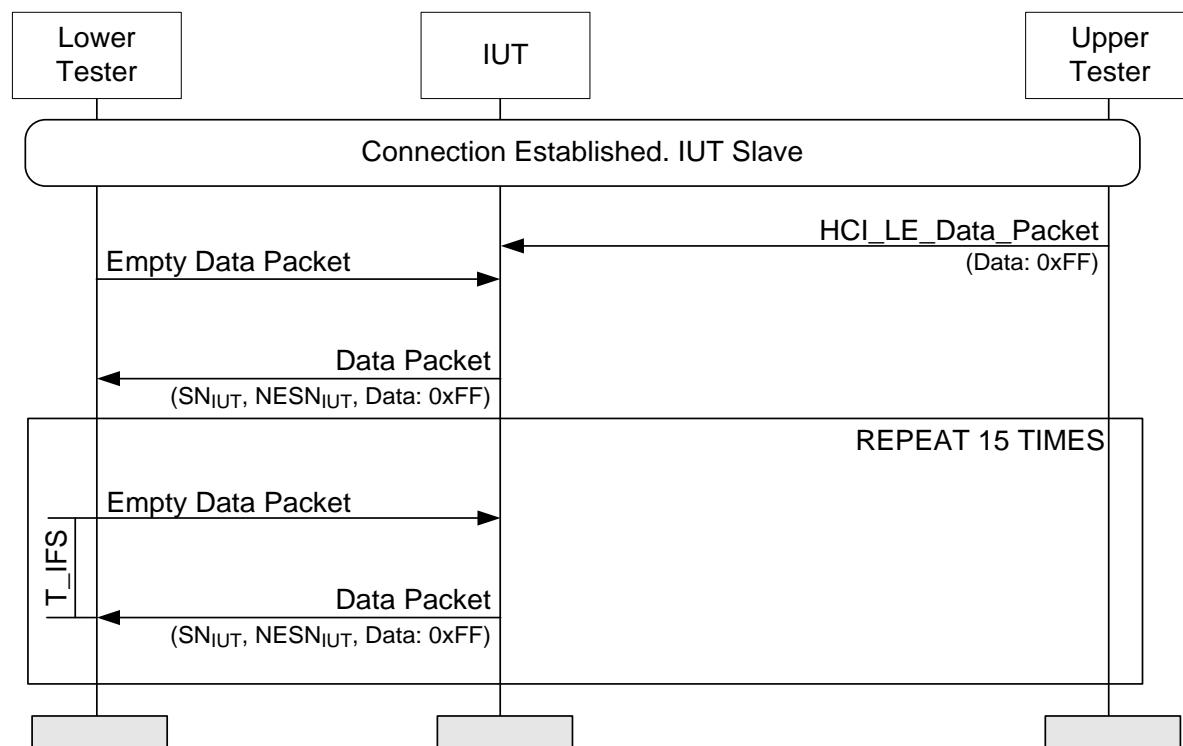


Figure 4.158: LL/CON/SLA/BV-16-C [Slave retransmission]

1. Lower Tester sends an Empty DATA packet once a connection interval to the IUT using the data channel selection parameters has been previously established. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet received is the next SN, but omit acknowledgements by using the current NESN.
 2. In events where the IUT is required to listen, the Lower Tester expects a DATA packet T_IFS after the packet sent. Expect packets received correctly from the IUT to retransmit the packet by the SN matching the current NESN and the NESN matching the next SN.
 3. Repeat steps 1-2 15 times.
 4. Lower Tester terminates the connection.

- Expected Outcome

Pass Verdict

The IUT retransmits the same SN and payload when asked for a retransmission.

4.3.4.12 LL/CON/SLA/BV-17-C [Slave Acknowledgement Repetition]

- Test Purpose

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and repeats a positive acknowledgement of a packet.

The Lower Tester acts in the master role, starting events using the default connection parameters, using negative acknowledgements to prompt a retransmission of an acknowledgement from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value).

- Test Procedure

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

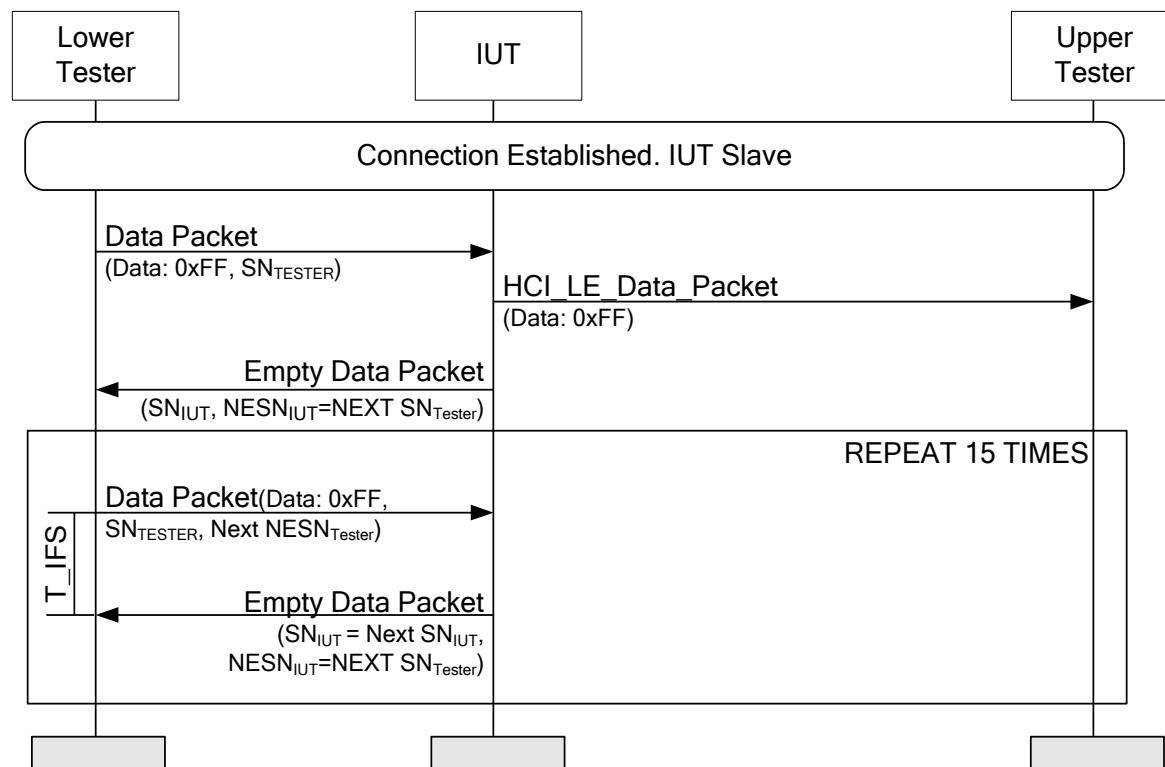


Figure 4.159: LL/CON/SLA/BV-17-C [Slave acknowledgement repetition]



1. Lower Tester sends a DATA packet (with SN_{TESTER}) once a connection interval to the IUT using the data channel selection parameters. Not to recognize an acknowledgement, use the current SN where in the previous packet received the NESN is the next SN, but acknowledge packets correctly by using the next NESN where the current NESN matches the SN in the previous packet correctly received.
 2. The IUT sends an acknowledgement (with SN_{IUT} and $NESN_{IUT} = \text{NEXT } SN_{TESTER}$).
 3. Lower Tester does not recognize the acknowledgement and resend the DATA packet using the SN_{TESTER} and payload equal to those of the previous packet sent by the Lower Tester.
 4. Lower Tester expects the IUT to repeat the acknowledgement ($SN_{IUT} = \text{NEXT } SN_{IUT}$, $NESN_{IUT} = \text{NEXT } SN_{TESTER}$).
 5. Repeat steps 3-4 15 times.
 6. Lower Tester terminates the connection.
- Expected Outcome

Pass Verdict

The IUT retransmits an acknowledgement when being retransmitted the same payload.

4.3.4.13 LL/CON/SLA/BV-18-C [Slave Lost Negative Acknowledgement]

- Test Purpose

Tests that a slave IUT is able to maintain a connection observing the acknowledgement scheme and preserve the packet sequence numbering in the case of a lost negative acknowledgement.

The Lower Tester acts in the master role, starting events using the default connection parameters, moving to the next packet after a negative acknowledgement to prompt a repeated negative acknowledgement from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.



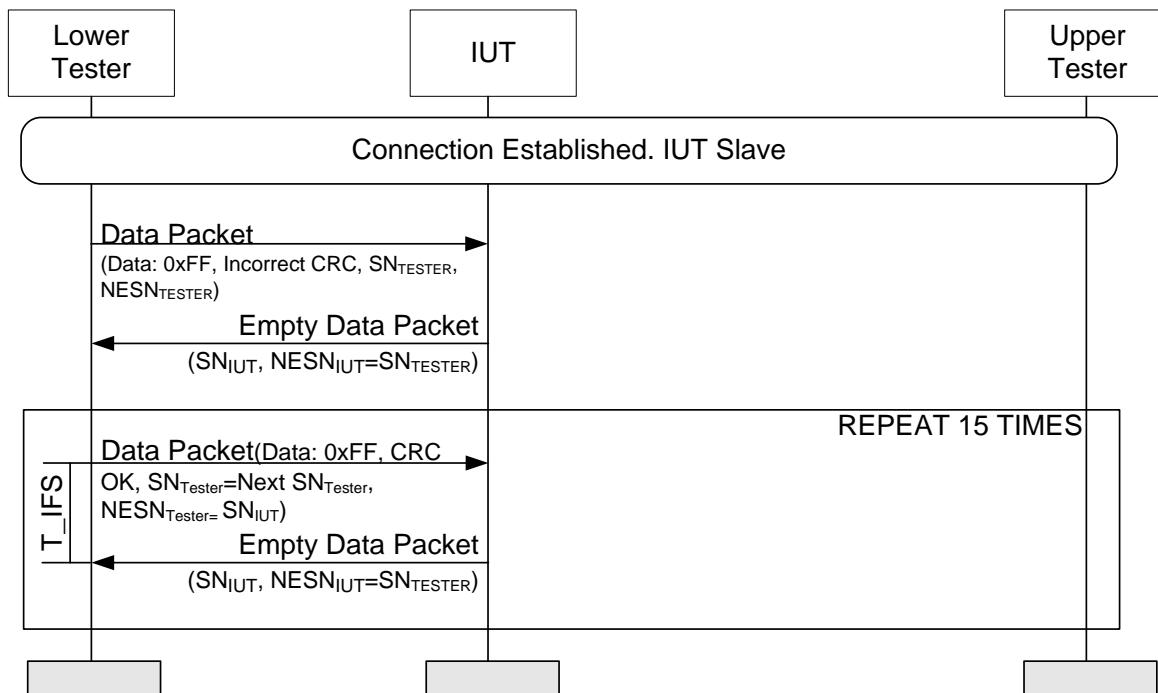


Figure 4.160: LL/CON/SLA/BV-18-C [Slave lost negative acknowledgement]

1. Lower Tester sends a DATA packet with a varying checksum once a connection interval to the IUT using the data channel selection parameters. To prompt the IUT to produce a negative acknowledgement, use the next SN for one packet to send with an invalid checksum, where the NESN in the previous packet received is the next SN, then if the following packet's NESN is the current SN use the next SN with a valid checksum. The Lower Tester responds with a negative acknowledgement to the IUT's packet by using a NESN that matches the SN of the IUT's packet.
 2. In events where the IUT is required to listen, Lower Tester expects a DATA packet T_IFS after the packet sent. Expect the IUT to repeat the packet, with the SN matching the current NESN and the NESN matching the current SN.
 3. Repeat steps 1-2 15 times.
 4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The IUT retransmits a negative acknowledgement when being transmitted the next payload.

4.3.4.14 LL/CON/SLA/BV-19-C [Slave Request Version]

- Test Purpose

Test that a connected slave IUT requests and performs the Version Exchange procedure.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to perform version exchange.

- Reference

[3] 5.1.5



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

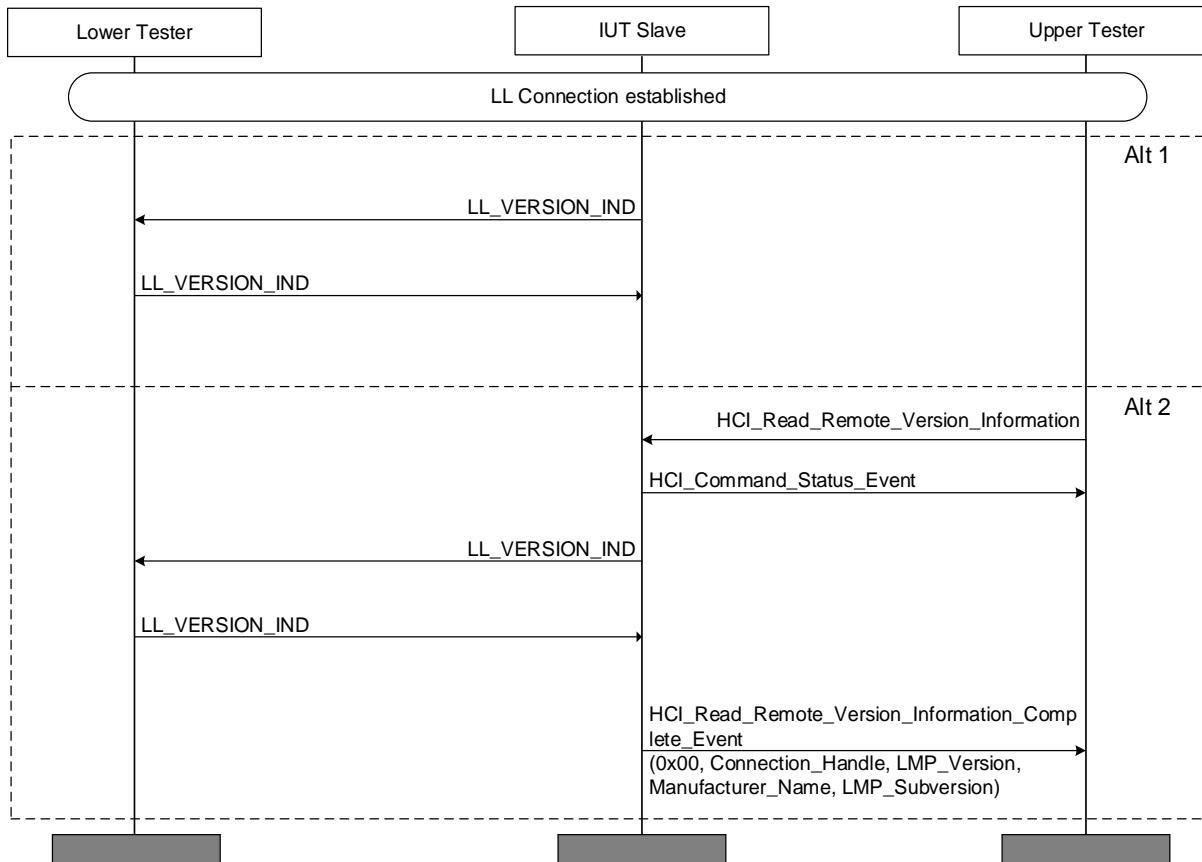


Figure 4.161: LL/CON/SLA/BV-19-C [Slave Request Version]

- Once connection is established, if the IUT has sent the Lower Tester a Version_Ind packet containing IUT version information by itself, the Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
 - If the IUT does not send the Lower Tester a Version_Ind packet containing IUT version information by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command and receives an HCI_Command_Status_Event.
 - The IUT sends to the Lower Tester a Version_Ind packet containing IUT version information.
 - The Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
 - The IUT sends to the Upper Tester the event HCI_Read_Remote_Version_Information_Complete_Event with Tester version information.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT requesting the version information and acknowledging the reply.



If the procedure was initiated by the Upper Tester as described in Step 2, the IUT reports the version requested completed with an HCI event.

4.3.4.15 LL/CON/SLA/BV-20-C [Slave Respond Version]

- Test Purpose

Test that a connected slave IUT responds to the request from the Lower Tester to perform the version exchange procedure.

The Lower Tester acts in the master role in a maintained connection and requests to perform version exchange.

- Reference

[3] 5.1.5

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

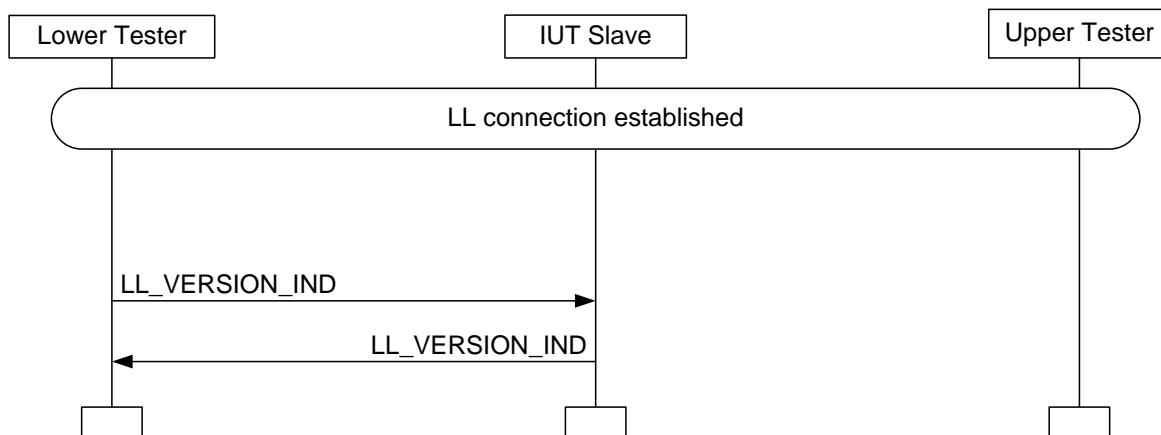


Figure 4.162: LL/CON/SLA/BV-20-C [Slave Respond Version]

1. Configure the Lower Tester to send Version_Ind packet
 2. Lower Tester sends a Version_Ind packet containing Tester version information.
 3. IUT acknowledges the Version_Ind packet and sends a Version_Ind packet containing IUT version information to the Lower Tester.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT responding to the version information.

Inconclusive Verdict

The IUT sends an LL_VERSION_IND PDU before the Lower Tester does.

4.3.4.16 LL/CON/SLA/BV-21-C [Slave Acknowledgement Scheme]

- Test Purpose

Test that a slave IUT is able to maintain a connection observing the acknowledgement scheme.



The Lower Tester acts in the master role, maintaining the connection and checking that the IUT uses correctly the acknowledgement scheme.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common connection interval, slave latency and timeout parameters. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN, assigning from the next to the current variables conditional on the packet contents from the IUT.

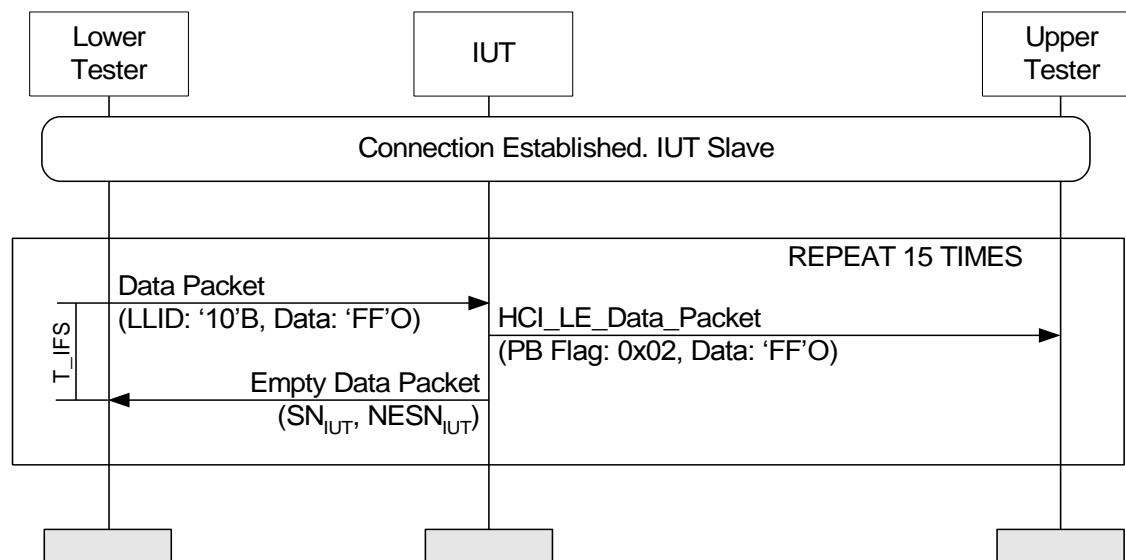


Figure 4.163: LL/CON/SLA/BV-21-C [Slave Acknowledgement Scheme]

1. Lower Tester sends a DATA packet once a connection interval to the IUT using the data channel selection parameters. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received matches the next SN and by using the next NESN where the SN in the previous packet correctly received matches the current NESN.
 2. In events where the IUT is required to listen, Lower Tester expects a DATA packet T_IFS after the packet sent, with the SN matching the current NESN and the NESN matching the next SN.
 3. Repeat steps 1-2 15 times.
 4. State: Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT using the normal acknowledgement scheme operation.



The IUT ACKs at least to 10 out of 15 packets sent by the Lower Tester.

4.3.4.17 LL/CON/SLA/BV-22-C [Initiate Feature Exchange]

- Test Purpose

Test that a connected slave IUT requests and performs the feature exchange procedure, activating the correct features.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to perform feature exchange.

- Reference

[3] 5.1.4.2

- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

- Test Procedure

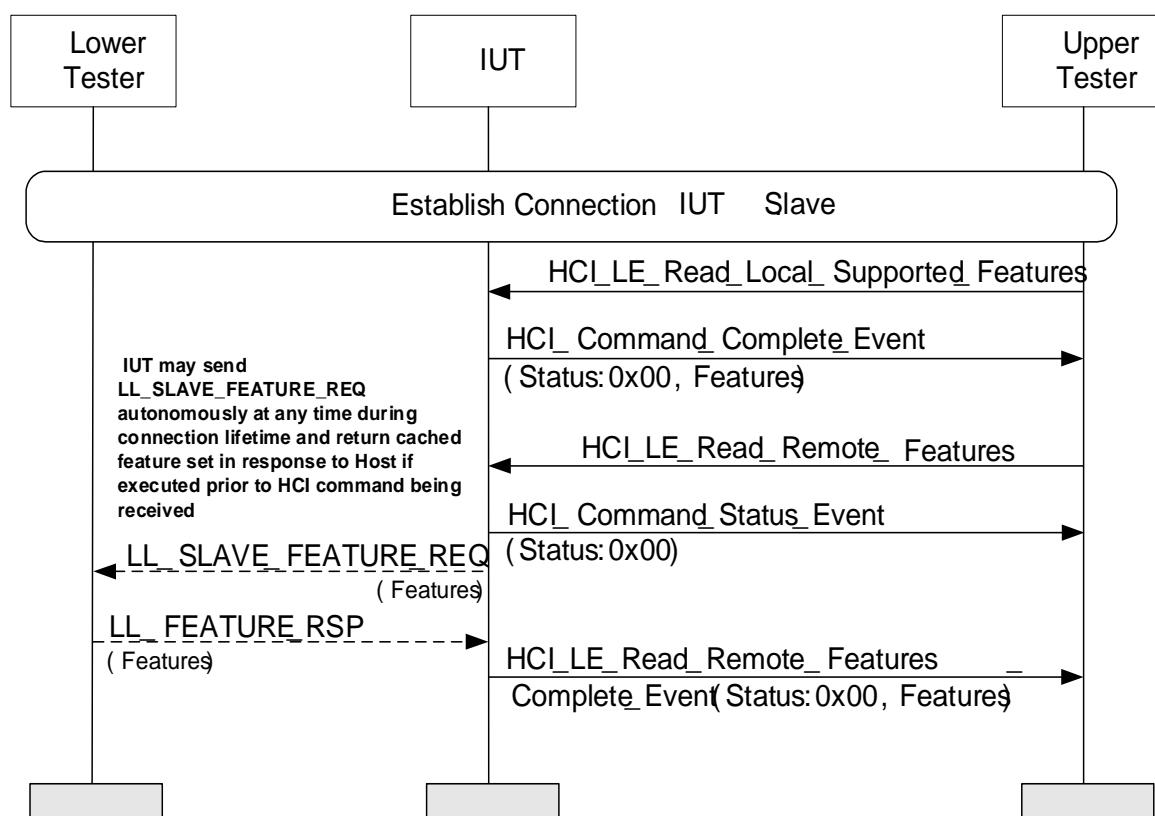


Figure 4.164: LL/CON/SLA/BV-22-C [Initiate Feature Exchange]

- Upper Tester sends an HCI_LE_Read_Local_Supported_Features command and receives an HCI_Command_Complete event with the correct feature set value.



2. Upper Tester sends an HCI_LE_Read_Remote_Features command and receives an HCI_Command_Status event.
 3. The IUT may have autonomously sent an LL_SLAVE_FEATURE_REQ prior to receiving the HCI command from the Upper Tester; otherwise, the IUT sends an LL_SLAVE_FEATURE_REQ PDU, including the configured feature set and waits for an LL FEATURE_RSP PDU from the Lower Tester.
 4. The IUT sends the HCI_LE_Read_Remote_Features_Complete event to the Upper Tester.
- Expected Outcome

Pass Verdict

All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT requesting the feature information and acknowledging the reply.

The IUT reports the feature exchange procedure completed with an HCI_LE_Read_Remote_Features_Complete event.

4.3.4.18 LL/CON/SLA/BV-23-C [Initiate Feature Exchange – Master does not support]

- Test Purpose

Test that a slave IUT requests and performs the feature exchange procedure when the master does not support that procedure.

The Lower Tester acts in the master role in the connection and responds to the request from the IUT with an LL_UNKNOWN_RSP.

- Reference
[3] 5.1.4.2
- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

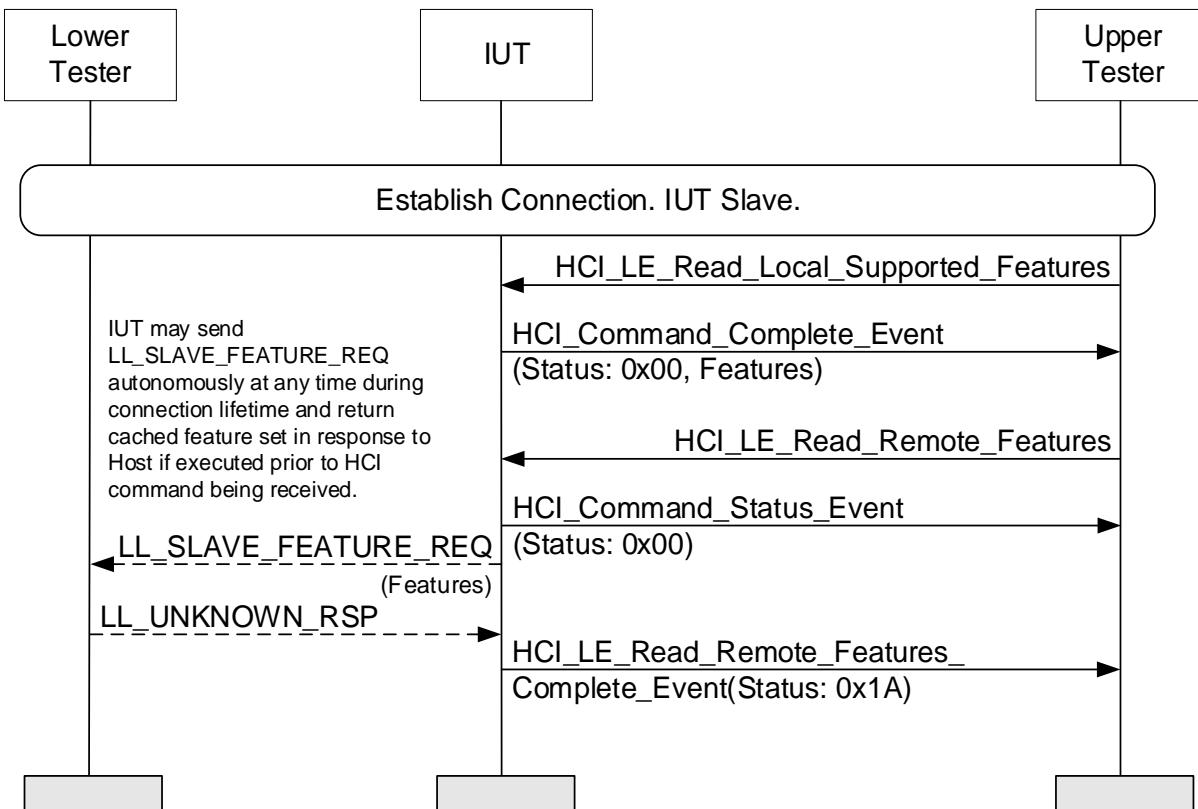


Figure 4.165: LL/CON/SLA/BV-23-C [Initiate Feature Exchange – Master does not support]

1. Upper Tester sends an **HCI_LE_Read_Remote_Feature** command and receives an **HCI_Command_Status_Event**.
 2. The IUT may have autonomously sent an **LL_SLAVE_FEATURE_REQ** prior to receiving the HCI command from the Upper Tester; otherwise, the IUT sends an **LL_SLAVE_FEATURE_REQ** PDU, including the configured feature set and waits for an **LL_FEATURE_RSP** PDU from the Lower Tester.
 3. Lower Tester acknowledges the **LL_SLAVE_FEATURE_REQ** PDU and sends an **LL_UNKNOWN_RSP** PDU in response.
 4. The IUT sends the **HCI_Le_Read_Remote_Feature_Complete** event to the Upper Tester with the correct error code.
- Expected Outcome

Pass Verdict

The IUT sends **HCI_Le_Read_Remote_Feature_Complete** (Reason: 0x1A) when the Lower Tester responds with an **LL_UNKNOWN_RSP** PDU.

4.3.4.19 LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device accepts the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower



Tester then observes the procedure carried out by the IUT and accepts the IUT's request. The actual parameters used by the IUT may be different from the parameters provided by the Upper Tester.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

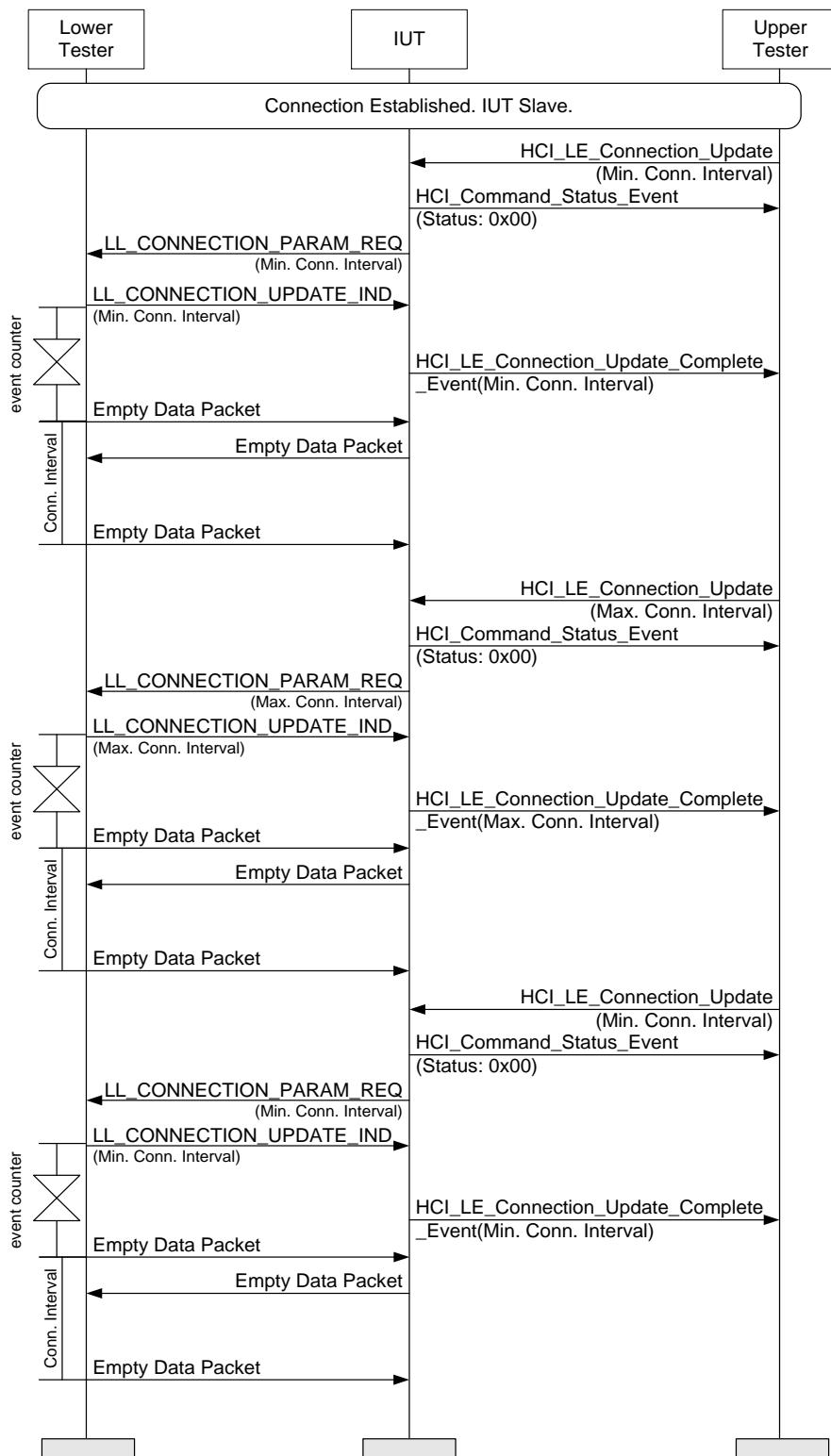


Figure 4.166: LL/CON/SLA/BV-24-C [Initiating Connection Parameter Request – Accept]



Case 1:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the maximum connection interval, no latency and maximum connection supervision timeout and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters selected by the IUT.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.



- Expected Outcome

Pass Verdict

For all three cases described in the test procedure all the following three conditions shall occur:

- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and uses the new parameters selected by the IUT at the assigned event.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.20 LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device rejects the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT's request.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
 LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than

LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

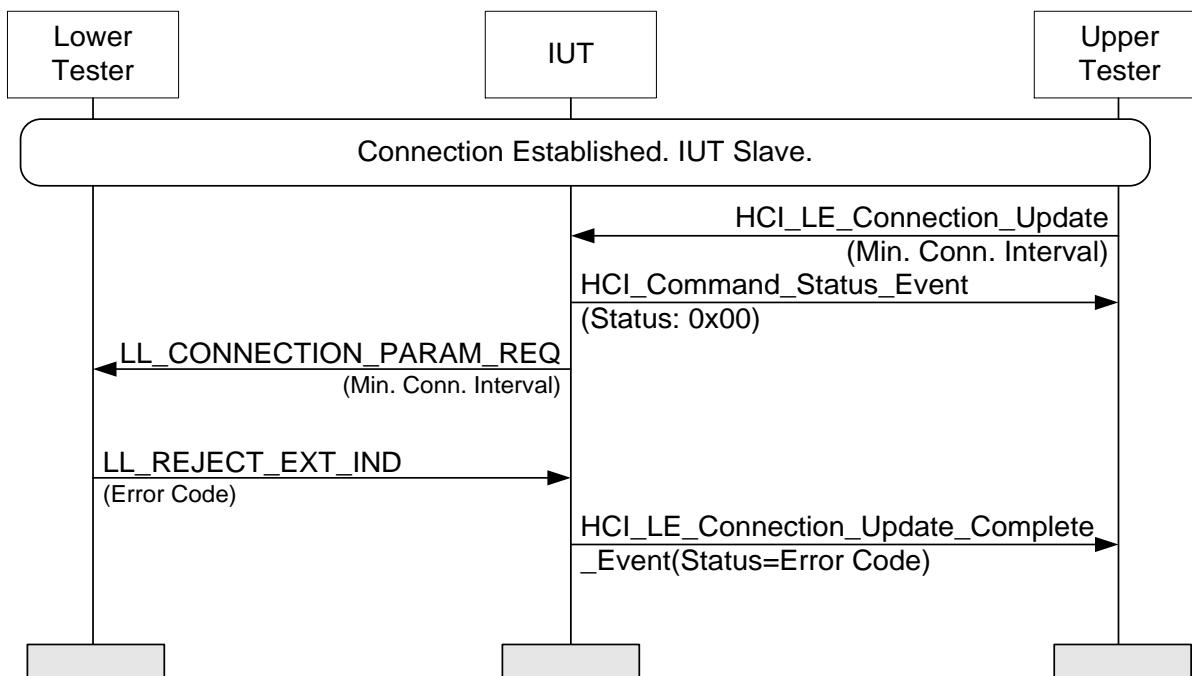


Figure 4.167: LL/CON/SLA/BV-25-C [Initiating Connection Parameter Request – Reject]

1. Upper Tester sends an **HCI LE Connection Update** command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an **HCI Command Status** event from the IUT in response.
2. Lower Tester expects an **LL CONNECTION PARAM REQ** control PDU from the IUT (the actual parameters in the **LL CONNECTION PARAM REQ** PDU may be different from the parameters provided by the Upper Tester) and the Lower Tester rejects the IUT's request by issuing an **LL REJECT EXT IND** PDU. Lower Tester expects a packet from the IUT acknowledging the **LL REJECT EXT IND** PDU.
3. Upper Tester expects an **HCI LE Connection Update Complete** from the IUT containing the error code sent by the Lower Tester.

- Expected Outcome

Pass Verdict

The IUT transmits the **LL CONNECTION PARAM REQ** PDU to update connection parameters.

The IUT reports an **HCI LE Connection Update Complete** event containing the error code sent by the Lower Tester.

4.3.4.21 LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT's connection parameter request and the Lower Tester's connection parameter request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower



Tester then observes the procedure carried out by the IUT and initiates a new connection parameter request procedure upon receiving the IUT's connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master's connection parameter request procedure after the master has rejected the IUT's connection parameter request procedure.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

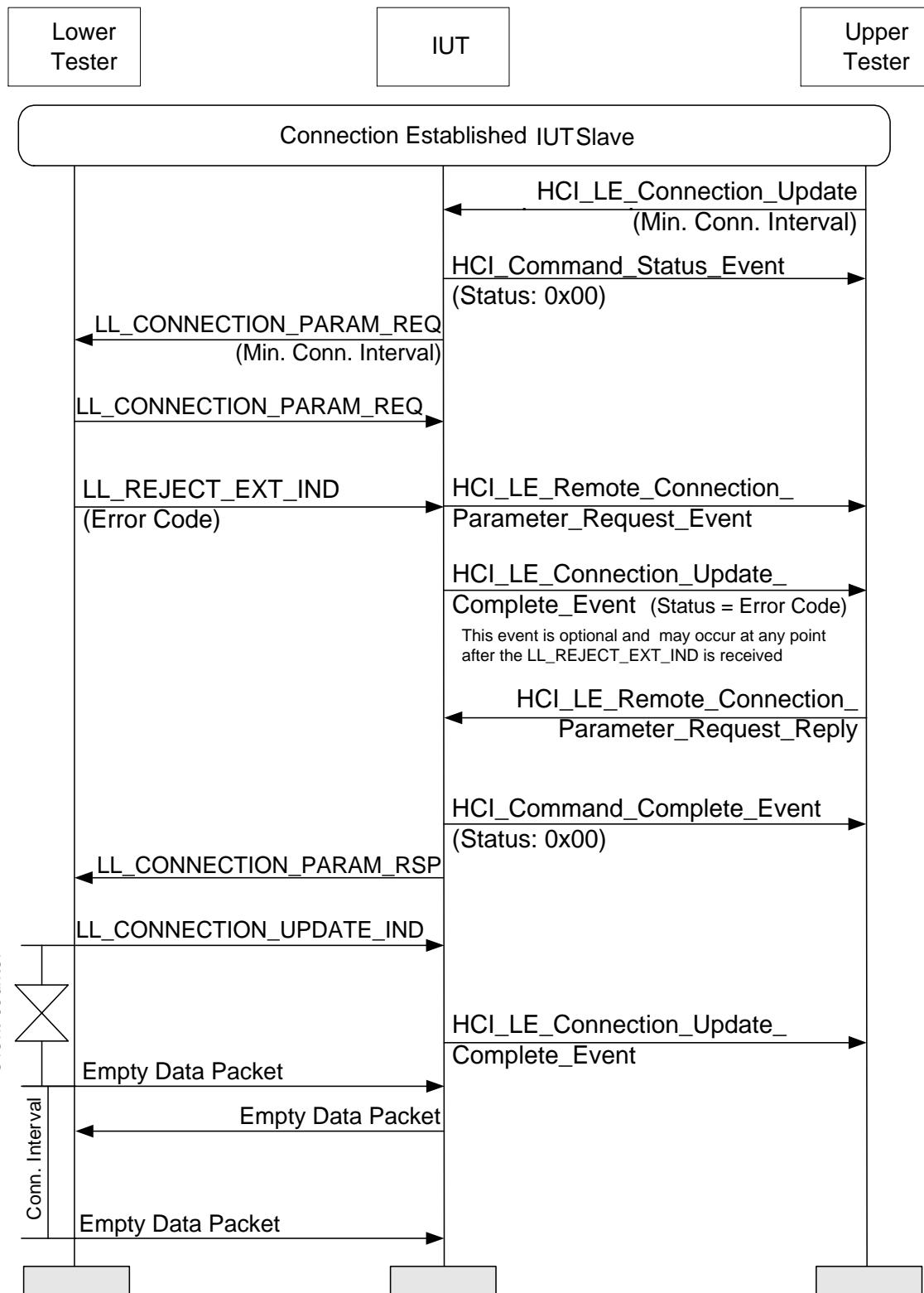


Figure 4.168: LL/CON/SLA/BV-26-C [Initiating Connection Parameter Request – same procedure collision]

1. Upper Tester sends an **HCI_LE_Connection_Update** command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an **HCI_Command_Status** event from the IUT in response.



2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_CONNECTION_PARAM_REQ and rejects the IUT's LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x23.
 3. The IUT requests the Upper Tester to accept or reject the Lower Tester's request.
 4. At this point or later in the procedure, the IUT may optionally send an HCI_LE_Connection_Update_Complete event with reason code set to 0x23 to the Upper Tester.
 5. The Upper Tester accepts the IUT's request to accept or reject the Lower Tester's request.
 6. The IUT responds to the LL_CONNECTION_PARAM_REQ from the Lower Tester with an LL_CONNECTION_PARAM_RSP.
 7. The Lower Tester responds with an LL_CONNECTION_UPDATE_IND PDU and expects a packet from the IUT acknowledging the connection update request.
 8. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
 9. At the time of the update start maintaining the connection with the new parameters.
 10. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome
- Pass Verdict
- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
 - If the IUT reports an HCI_LE_Connection_Update_Complete event when the Lower Tester rejects the IUT's request, it shall have the correct error code (0x23).
 - The IUT responds positively to the master initiated connection parameter request procedure.
 - The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event.
 - The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.22 LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT's connection parameter request and the remote device's channel map update.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a channel map update procedure upon receiving the IUT's connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master's channel map update procedure after the master has rejected the IUT's connection parameter request procedure.

- Reference

[3] 5.1.7



- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

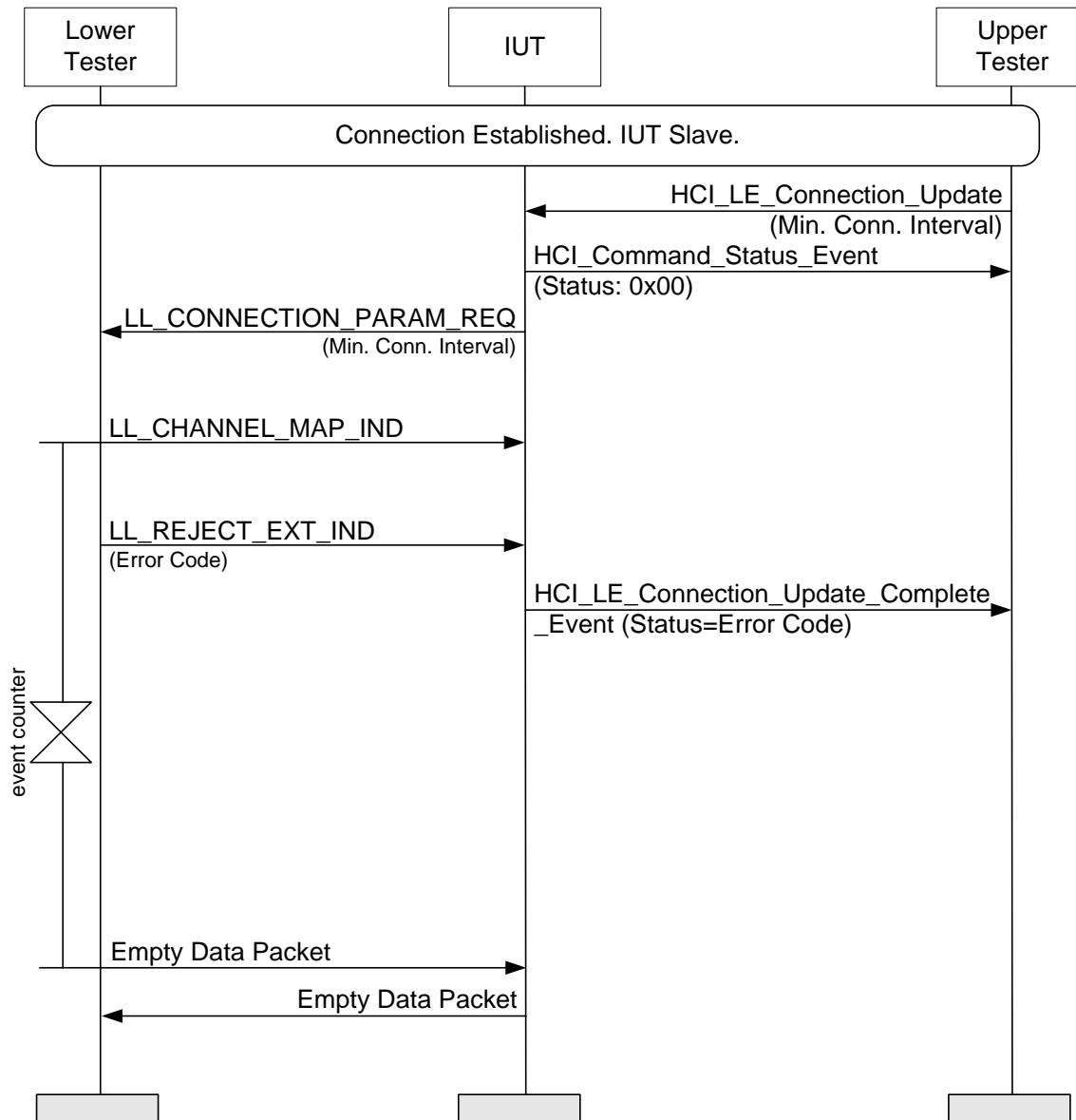


Figure 4.169: LL/CON/SLA/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

- Upper Tester sends an HCI LE Connection Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI Command Status event from the IUT in response.



2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_CHANNEL_MAP_IND and rejects the IUT's LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND.
 3. IUT sends the HCI_LE_Connection_Update_Complete event with reason code set to 0x2A to the Upper Tester.
 4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of channel map update.
- Expected Outcome
- Pass Verdict
- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and reports an HCI_LE_Connection_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT's request.
 - The IUT responds positively to the master initiated channel map request procedure.
 - The test procedure is executed successfully, with the IUT acknowledging the channel map request and adopting the new channel map at the assigned event.
 - The IUT maintains the connection with the Lower Tester after the channel map update completes.

4.3.4.23 LL/CON/SLA/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT's connection parameter request and the remote device's encryption procedure.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates an encryption start procedure upon receiving the IUT's connection parameter request to cause a procedure collision. The test case expects the IUT to respond to the master's encryption start procedure and then complete the IUT's connection parameter request procedure.

- Reference
- [3] 5.1.7, 5.1.3.1
- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

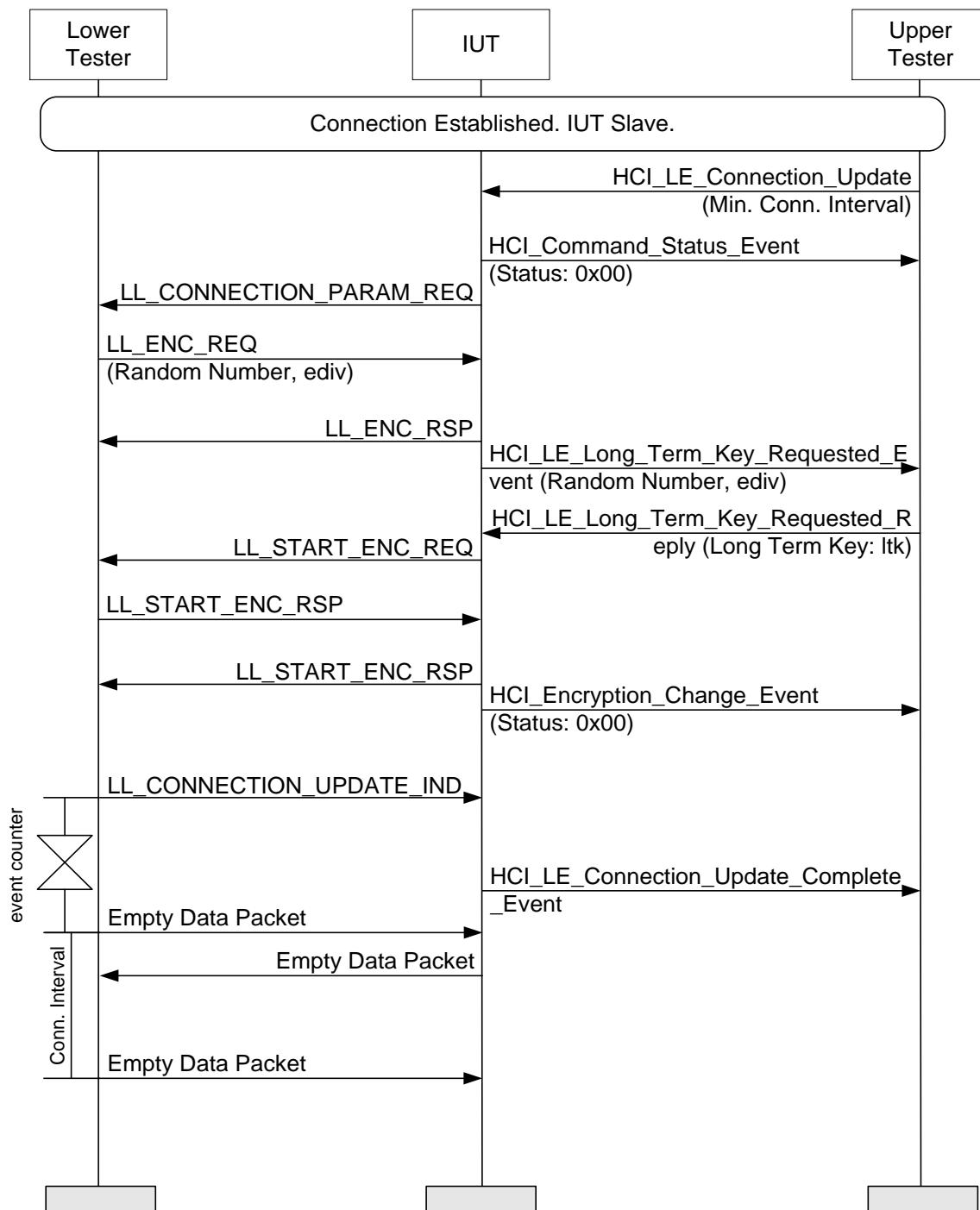


Figure 4.170: LL/CON/SLA/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

- Upper Tester sends an **HCI_LE_Connection_Update** command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an **HCI_Command_Status** event from the IUT in response.
- Lower Tester expects an **LL_CONNECTION_PARAM_REQ** control PDU from the IUT (the actual parameters in the **LL_CONNECTION_PARAM_REQ** PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an **LL_ENC_REQ** PDU.

3. The Lower Tester expects the IUT to respond with an LL_ENC_RSP.
 4. The Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT.
 5. The Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply to the IUT and expects an HCI_Command_Complete in response.
 6. The Lower Tester expects an LL_START_ENC_REQ from the IUT which it acknowledges.
 7. The Lower Tester sends an LL_START_ENC_RSP PDU to the IUT and expects an LL_START_ENC_RSP PDU from the IUT in response.
 8. The Upper Tester expects an HCI_Encryption_Change event with encryption enable set to on.
 9. The Lower Tester responds to the IUT's connection parameter request with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
 10. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
 11. At the time of the update start maintaining the connection with the new parameters.
 12. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome
- Pass Verdict
- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
 - The IUT sends its initialization vector and session key diversifier in a LL_ENC_RSP packet
 - The IUT reports the encryption setup requested with the HCI event HCI_LE_Long_Term_Key_Requested.
 - The IUT sends a LL_START_ENC_REQ packet until acknowledged.
 - The IUT acknowledges the LL_START_ENC_RSP and responds with one.
 - The IUT successfully reports the encryption change with the HCI event HCI_Encryption_Change.
 - The IUT acknowledges the connection update request from the Lower Tester and adopts the new parameters selected by the IUT at the assigned event.
 - The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.
- Notes
- The Lower Tester and Upper Tester ensure that the encryption start procedure completes before the procedure response timeout for the connection parameter request procedure fires.

4.3.4.24 LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity]

- Test Purpose
- Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master does not indicate any preferred periodicity.
- Reference
- [3] 5.1.7



- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

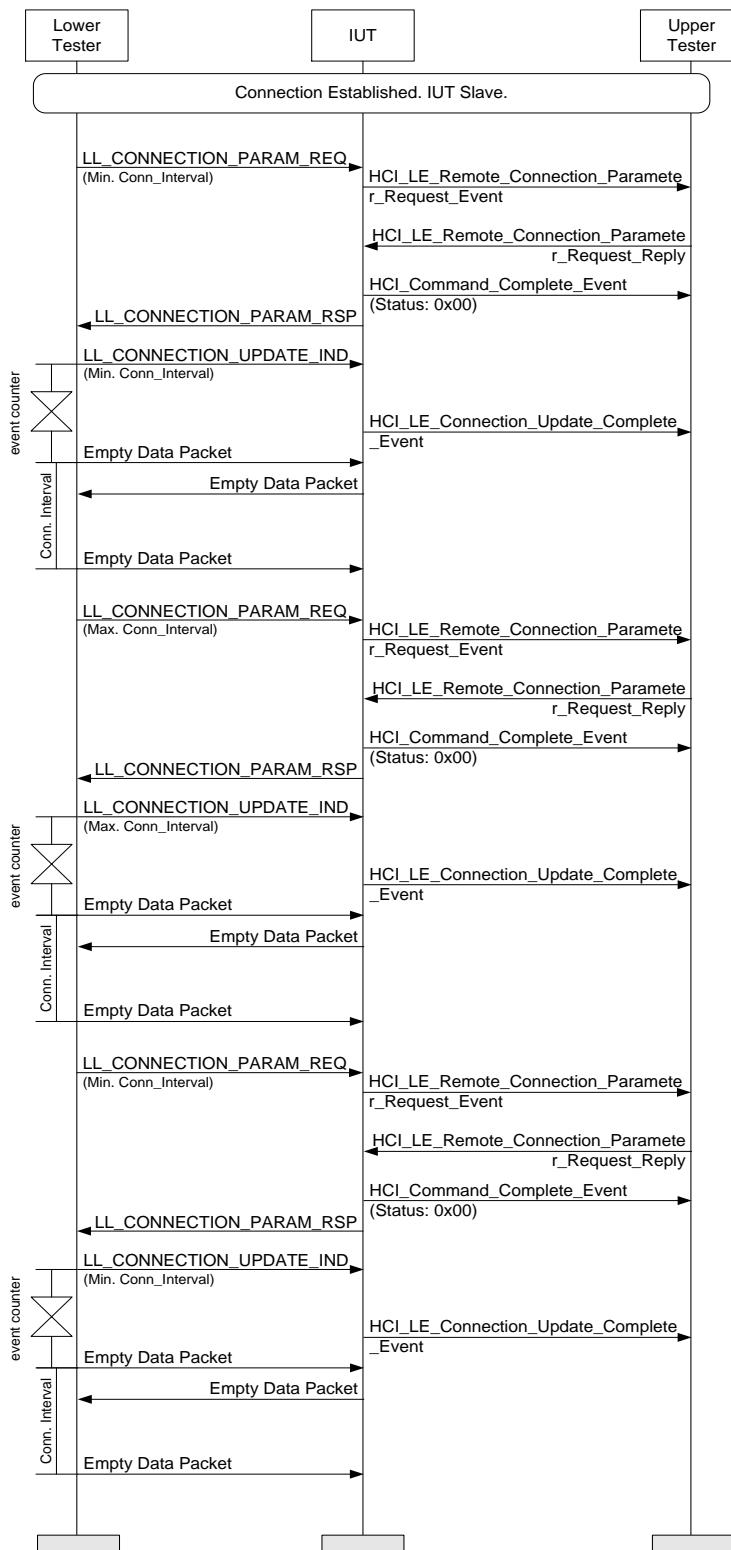


Figure 4.171: LL/CON/SLA/BV-29-C [Accepting Connection Parameter Request – no Preferred_Periodicity]



Case 1:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and PreferredPeriodicity set to zero.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the maximum connection interval, no latency, and maximum connection supervision timeout and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.



- Expected Outcome

Pass Verdict

For all three cases described in the test procedure, the following conditions shall occur:

- The IUT responds positively to the Lower Tester's request to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.25 LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master only requests a change in anchor points.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

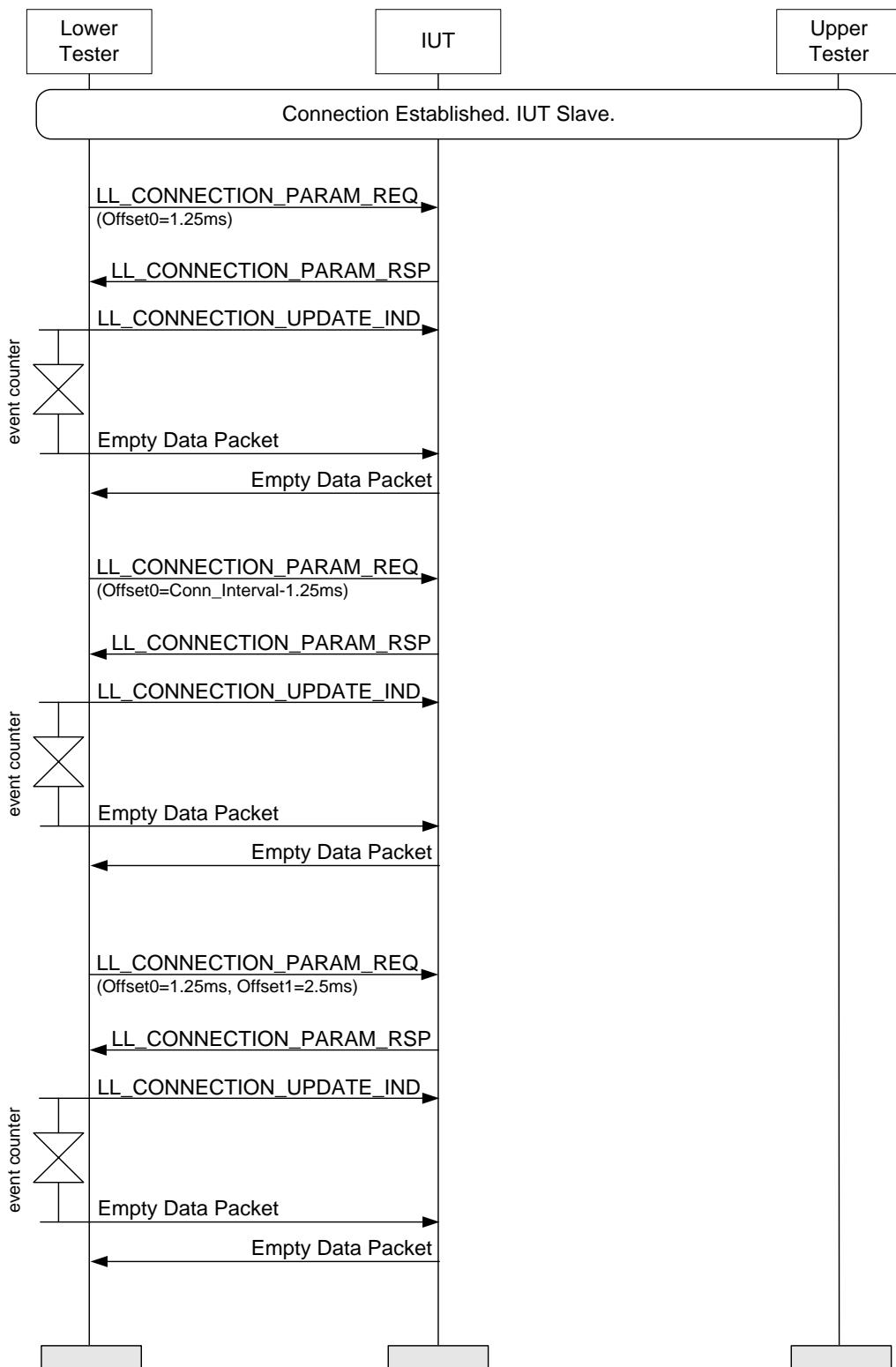


Figure 4.172: LL/CON/SLA/BV-30-C [Accepting Connection Parameter Request – preferred anchor points only]



Case 1:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are 1.25ms from the old anchor points. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to (connection interval – 1.25ms), Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are (connection interval - 1.25ms) from the old anchor points. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to 2.5ms, Offset2-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT and responds with an LL_CONNECTION_UPDATE_IND PDU such that the new anchor points are 1.25ms from the old anchor points. Lower Tester expects a packet from the IUT acknowledging the connection update request.
3. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
4. At the time of the update start maintaining the connection with the new parameters.

- Expected Outcome

Pass Verdict

- The IUT responds positively to the Lower Tester's request to update connection parameters.
- The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters at the assigned event:
 - In the first case, the anchor points are shifted by 1.25ms.
 - In the second case, the anchor points are shifted by (connection interval - 1.25ms).
 - In the third case, the anchor points are shifted by 1.25ms.



4.3.4.26 LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master indicates a preferred periodicity.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

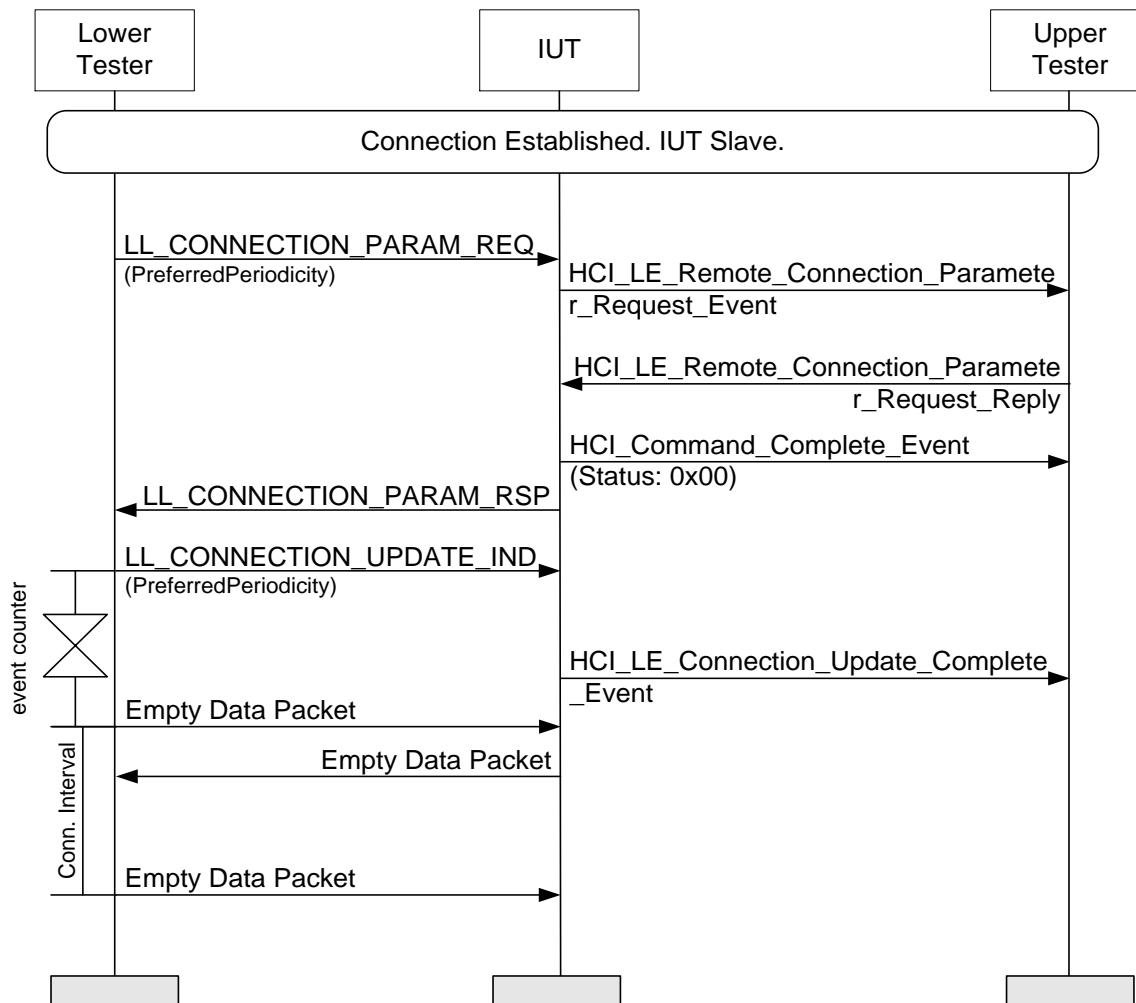


Figure 4.173: LL/CON/SLA/BV-31-C [Accepting Connection Parameter Request – Preferred_Periodicity]



1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s) and a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range.
 2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
 3. Lower Tester expects an LL_CONNECTION_PARAM_RSP control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_RSP PDU may be different from the parameters provided by the Upper Tester) and responds with an LL_CONNECTION_UPDATE_IND PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
 4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
 5. At the time of the update start maintaining the connection with the new parameters selected by the Lower Tester.
 6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester's request to update connection parameters.

The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.4.27 LL/CON/SLA/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master indicates a preferred periodicity and preferred anchor points.

- Reference

[3] 5.1.7

- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

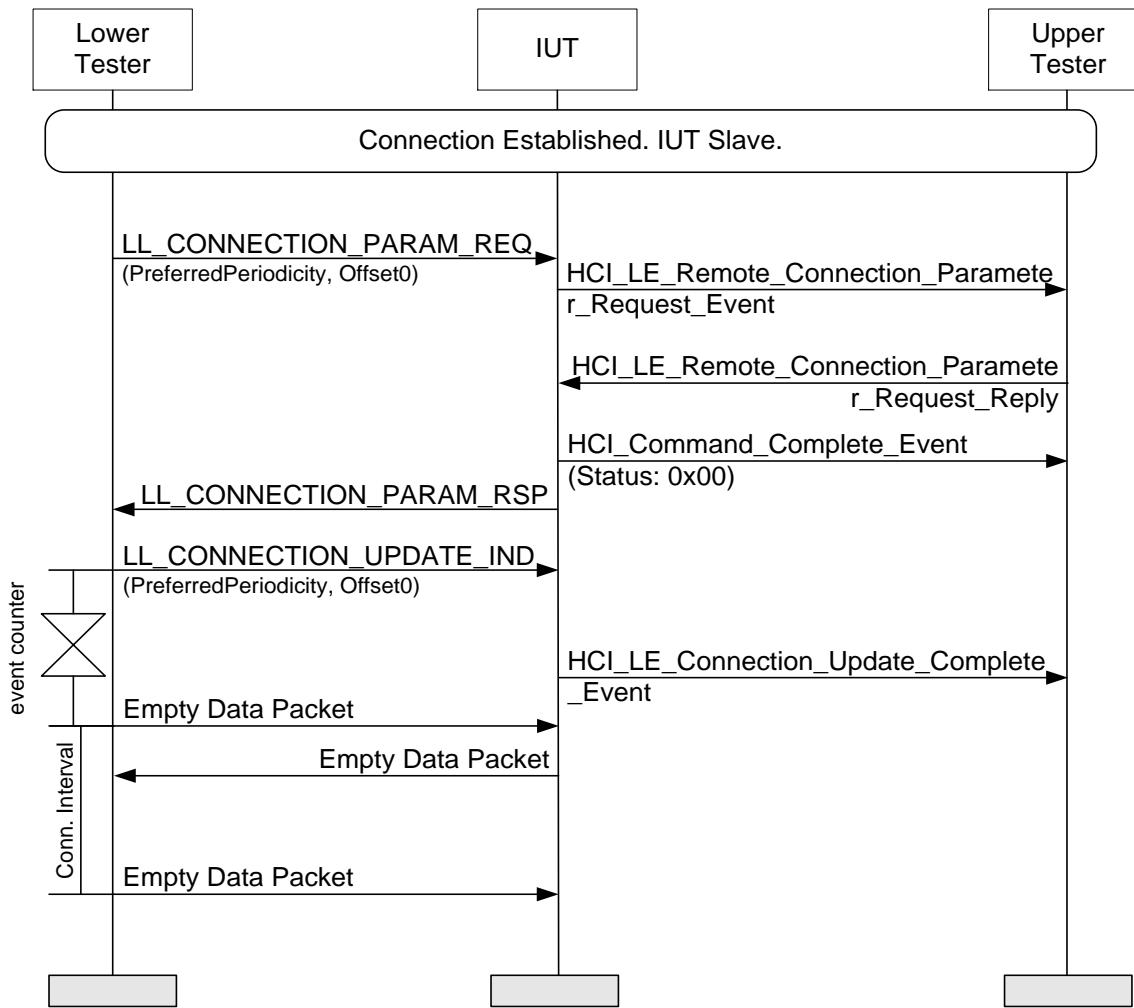


Figure 4.174: LL/CON/SLA/BV-32-C [Accepting Connection Parameter Request –Preferred_Periodicity and preferred anchor points]

1. Lower Tester sends an **LL_CONNECTION_PARAM_REQ** PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s), a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range, a reference connection event counter and a valid Offset0 value such that the new connection event is 1.25ms away from the old connection event at the reference connection event count (Offset1-5 are invalid).
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an **LL_CONNECTION_PARAM_RSP** control PDU from the IUT (the actual parameters in the **LL_CONNECTION_PARAM_RSP** PDU may be different from the parameters provided by the Upper Tester) and responds with an **LL_CONNECTION_UPDATE_IND** PDU. Lower Tester expects a packet from the IUT acknowledging the connection update request.
4. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
5. At the time of the update the IUT starts maintaining the connection with the new parameters selected by the Lower Tester.

6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester's request to update connection parameters.

The test procedure is executed successfully, with the IUT acknowledging the connection update request and adopting the new parameters selected by the Lower Tester at the assigned event,

The IUT reports the selected new connection parameters with an
HCI_LE_Connection_Update_Complete event.

4.3.4.28 LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master requires the slave LL to request for approval from the slave's Host and the slave's Host has masked the LE Remote Connection Parameter Request Event.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to
LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

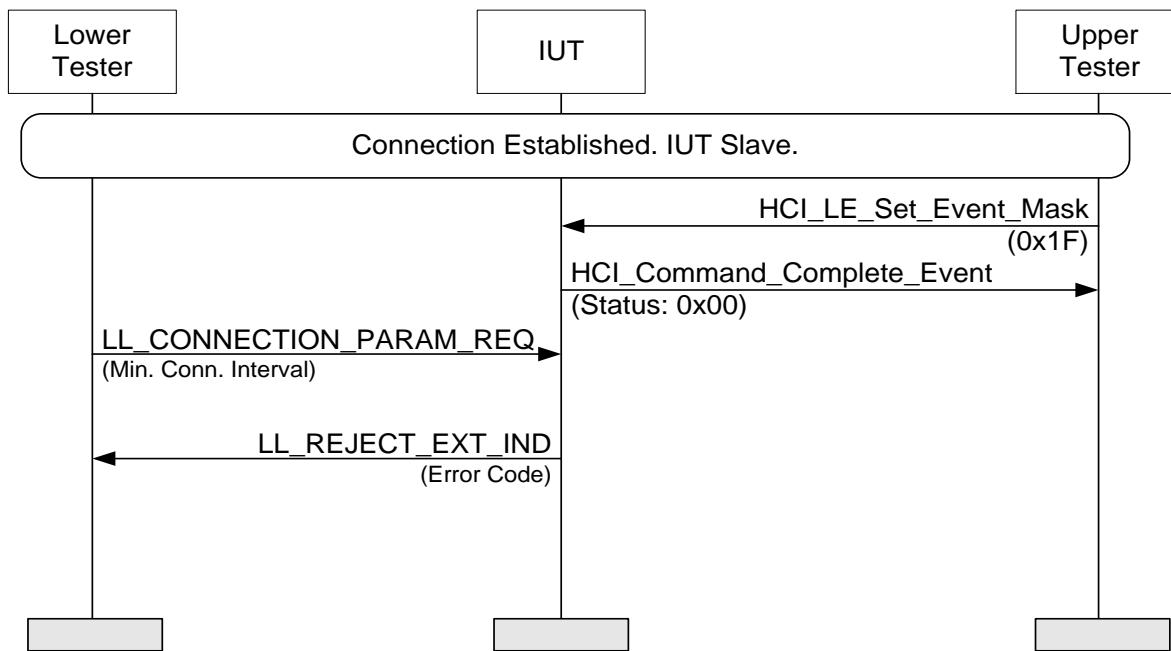


Figure 4.175: LL/CON/SLA/BV-33-C [Accepting Connection Parameter Request – event masked]

1. Upper Tester masks the LE Remote Connection Parameter Request event on the IUT.
2. Lower Tester sends an **LL_CONNECTION_PARAM_REQ** PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
3. Lower Tester expects an **LL_REJECT_EXT_IND** control PDU from the IUT with ErrorCode 0x1A.

- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an **LL_REJECT_EXT_IND** using the correct ErrorCode (0x1A).

4.3.4.29 LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the slave's Host rejects the master's connection parameter request procedure.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: **LL_slave_connInterval_MIN**, **LL_slave_connInterval_MAX**, **LL_slave_connSlaveLatency_MIN**, **LL_slave_connSlaveLatency_MAX**, **LL_connTimeout_MIN**, **LL_connTimeout_MAX**.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than **LL_slave_connSlaveLatency_MIN** up to **LL_slave_connSlaveLatency_MAX**, selected timeout, any SCA value).



- Test Procedure

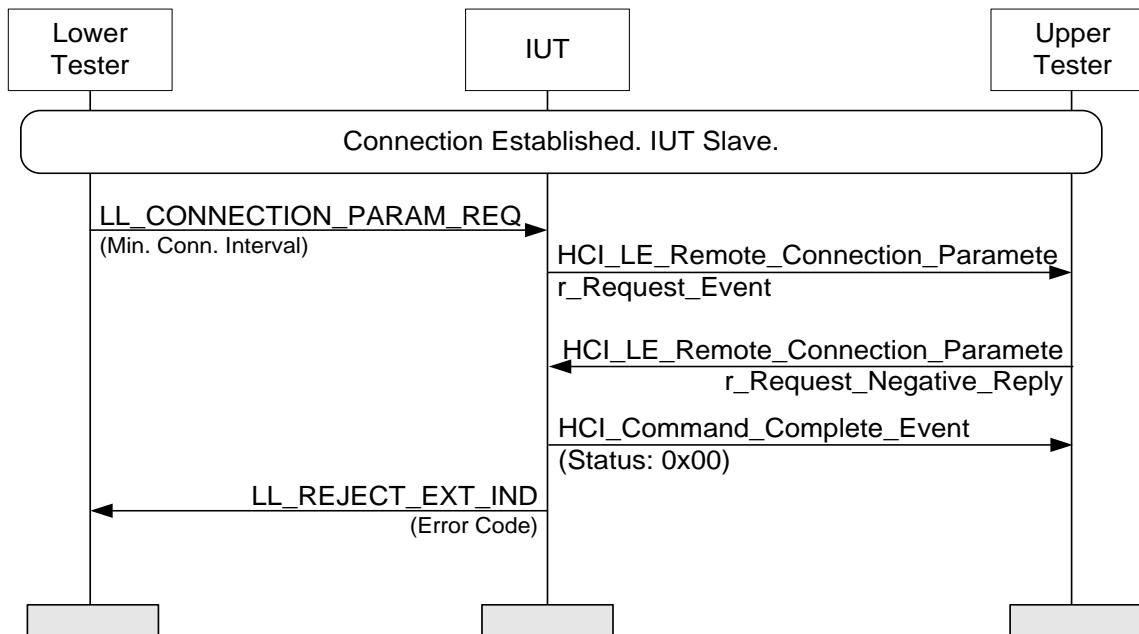


Figure 4.176: LL/CON/SLA/BV-34-C [Accepting Connection Parameter Request – Host rejects]

1. Lower Tester sends an **LL_CONNECTION_PARAM_REQ** PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester rejects the request using ErrorCode 0x3B.
3. Lower Tester expects an **LL_REJECT_EXT_IND** control PDU from the IUT containing the ErrorCode provided by the Upper Tester.

- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an **LL_REJECT_EXT_IND** using the ErrorCode provided by the Upper Tester.

4.3.4.30 LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and accepts the IUT's request.

- Reference

[10] 5.1.10



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

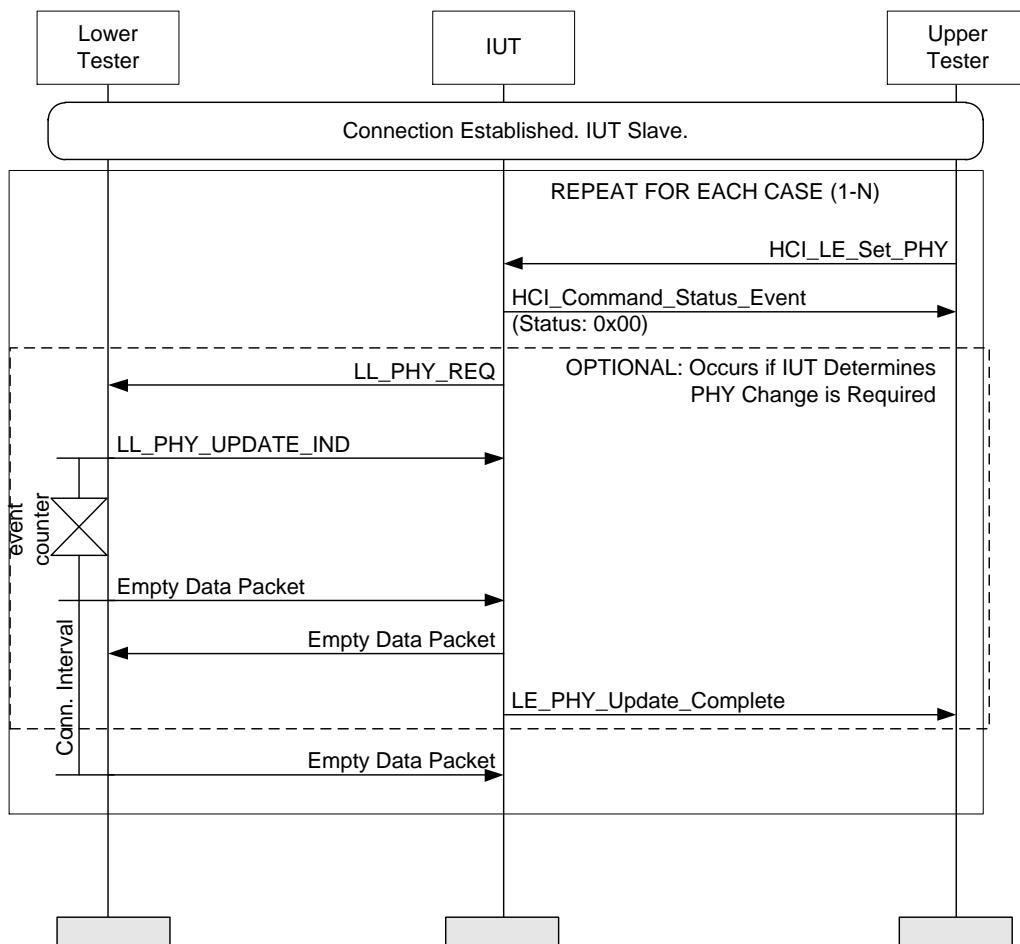


Figure 4.177: LL/CON/SLA/BV-40-C [Initiating PHY Update Procedure]

The following steps shall be carried out 2N times as follows, where N is the number of cases in Table 4.33:

- firstly using cases 1 to N from the relevant table in order;
 - then using the cases from the relevant table in a random order.
- Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the payload defined in the HCI_LE_Set_PHY section of the relevant table and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
 - If any bits set in TX_PHYS or RX_PHYS correspond to unsupported PHYs, the command shall be rejected by the IUT which shall return the error code Unsupported Feature or Parameter Value (0x11) to the Upper Tester. Following that proceed to the next round, rather than step 9, in step 3 if the IUT does NOT initiate a PHY change. Otherwise the IUT shall accept the command.



3. If the IUT initiates a PHY change, Lower Tester expects an LL_PHY_REQ control PDU from the IUT with at least one bit in each field (TX_PHYS, RX_PHYS) set. Proceed to step 4).
- OR
- If the IUT does NOT initiate a PHY change, proceed to step 9).
4. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the values defined in the "Lower Tester preference" section of the relevant table bitwise ANDed against the value sent by the IUT in the LL_PHY_REQ PDU such that:
 - a. M_TO_S_PHY = (M_TO_S_PHY_LTPREF) & (LL_PHY_REQ RX_PHYS Field)
 - b. S_TO_M_PHY = (S_TO_M_PHY_LTPREF) & (LL_PHY_REQ TX_PHYS Field)
 - c. If the IUT specifies the same single PHY in both the RX_PHYS and TX_PHYS fields, the Lower Tester shall use the PHY selected by the IUT for both directions or make no change. This rule shall take precedence over the values determined in rules (a) and (b).
 - d. If any of rules (a) to (c) result in a PHY equal to the current PHY, the corresponding field shall be set to zero rather than the bit corresponding to that PHY.
 - e. If either M_TO_S_PHY or S_TO_M_PHY are non-zero, then (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If M_TO_PHY and S_TO_M_PHY are both zero, the Instant shall be zero.
5. Lower Tester expects a packet from the IUT acknowledging the LL_PHY_UPDATE_IND.
6. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
7. At the Instant of the PHY change, start maintaining the connection with the new PHY(s) selected by the LL_PHY_UPDATE_IND PDU (or no change, if no change was specified in the LL_PHY_UPDATE_IND PDU).
8. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements. If PHY has changed, the Lower Tester shall use the new PHY.
9. If the command was accepted in step 2 or at least one of the PHY fields in the LL_PHY_UPDATE_IND PDU was non-zero, the Upper Tester expects LE_PHY_Update_Complete event from the IUT with a payload consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in the LL_PHY_UPDATE_IND PDU was zero or the LL_PHY_UPDATE_IND PDU was not sent). Otherwise the Upper Tester expects no event.

Case	HCI_LE_Set_PHY			Lower Tester preference	
	ALL_PHYS	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
1	0x00	0x01	0x03	0x02	0x00
2	0x00	0x02	0x05	0x01	0x02
3	0x00	0x04	0x02	0x02	0x04
4	0x00	0x02	0x01	0x00	0x02
5	0x00	0x01	0x04	0x04	0x00
6	0x00	0x06	0x03	0x02	0x02
7	0x00	0x01	0x01	0x01	0x01
8	0x00	0x03	0x04	0x00	0x01
9	0x00	0x01	0x05	0x00	0x00
10	0x00	0x02	0x04	0x04	0x02
11	0x00	0x07	0x03	0x01	0x00
12	0x00	0x06	0x06	0x02	0x04



Case	HCI_LE_Set_PHY			Lower Tester preference	
	ALL_PHYS	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
13	0x00	0x02	0x03	0x02	0x02
14	0x00	0x06	0x01	0x01	0x04
15	0x00	0x06	0x05	0x01	0x04
16	0x00	0x05	0x04	0x04	0x01
17	0x00	0x05	0x01	0x01	0x00
18	0x00	0x03	0x05	0x01	0x02
19	0x00	0x04	0x06	0x02	0x04
20	0x00	0x07	0x02	0x00	0x01
21	0x00	0x01	0x06	0x02	0x00
22	0x00	0x02	0x02	0x02	0x00
23	0x00	0x04	0x03	0x01	0x04
24	0x00	0x03	0x07	0x04	0x01
25	0x00	0x01	0x02	0x02	0x01
26	0x00	0x03	0x03	0x01	0x01
27	0x00	0x03	0x02	0x02	0x01
28	0x00	0x07	0x04	0x00	0x02
29	0x00	0x04	0x07	0x00	0x00
30	0x00	0x01	0x07	0x04	0x00
31	0x00	0x05	0x06	0x04	0x01
32	0x00	0x06	0x02	0x00	0x04
33	0x00	0x07	0x07	0x01	0x01
34	0x00	0x06	0x04	0x04	0x02
35	0x00	0x05	0x02	0x02	0x01
36	0x00	0x02	0x06	0x04	0x00
37	0x00	0x02	0x07	0x01	0x00
38	0x00	0x06	0x07	0x04	0x04
39	0x00	0x07	0x06	0x02	0x02
40	0x00	0x03	0x06	0x00	0x02
41	0x00	0x04	0x05	0x04	0x04
42	0x00	0x05	0x07	0x04	0x01
43	0x00	0x07	0x01	0x00	0x04
44	0x00	0x06	0x06	0x02	0x02
45	0x00	0x06	0x06	0x04	0x04
46	0x01	0x00	0x01	0x01	0x01
47	0x02	0x03	0x00	0x02	0x02
48	0x03	0x00	0x00	0x01	0x01



Case	HCI_LE_Set_PHY			Lower Tester preference	
	ALL_PHYS	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
49	0x02	0x05	0x00	0x04	0x04
50	0x03	0x00	0x00	0x04	0x04
51	0x03	0x00	0x00	0x02	0x02
52	0x01	0x00	0x06	0x04	0x04
53	0x01	0x00	0x06	0x04	0x02
54	0x02	0x07	0x00	0x02	0x01
55	0x02	0x06	0x00	0x02	0x02
56	0x01	0x00	0x08	0x02	0x02
57	0x01	0x00	0x11	0x01	0x00
58	0x01	0x00	0x20	0x02	0x01
59	0x01	0x00	0x41	0x01	0x01
60	0x01	0x00	0x80	0x01	0x02
61	0x01	0x00	0xFF	0x00	0x01
62	0x02	0x08	0x00	0x00	0x00
63	0x02	0x11	0x00	0x02	0x00
64	0x02	0x20	0x00	0x00	0x02
65	0x02	0x41	0x00	0x01	0x01
66	0x02	0x80	0x00	0x02	0x02
67	0x02	0xFF	0x00	0x01	0x01

Table 4.33: PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported.

- Expected Outcome

Pass Verdict

For all cases described in the test procedure all the following conditions shall occur:

- If the IUT transmits an LL_PHY_REQ PDU, at least one bit shall be set in each field (TX_PHYS, RX_PHYS).
- If the IUT is symmetric only, then it shall always select a single (symmetric) PHY in each LL_PHY_REQ PDU.
- If IUT transmits an LL_PHY_REQ, the change procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and using the PHY(s) selected by the LL_PHY_UPDATE_IND at the assigned event and thereafter.
- The IUT reports the currently selected PHY with a LE_PHY_Update_Complete event with the RX_PHY and TX_PHY fields consistent with the PHY(s) selected via the LL_PHY_UPDATE_IND PDU if either the command was accepted in step 2 or a PHY change occurred. If no change occurs, the RX_PHY and TX_PHY fields shall reflect the previous PHY still in use.
- The IUT does not send a LE_PHY_Update_Complete event if the command was rejected in step 2 and either the IUT did not initiate the PHY Update Procedure or it initiated the procedure but no PHY change occurred.



Fail Verdict

The IUT accepts the command in step 2 when a bit set in TX_PHYS or RX_PHYS corresponds to an unsupported PHY.

Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.4.31 LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure]

- **Test Purpose**

Test that a slave IUT is able to respond to a PHY update procedure from a master device. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the master role maintaining a connection and initiates the PHY update procedure. IUT responds to the PHY change request and notifies the host of the change only when appropriate.

- **Reference**

[10] 5.1.10

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

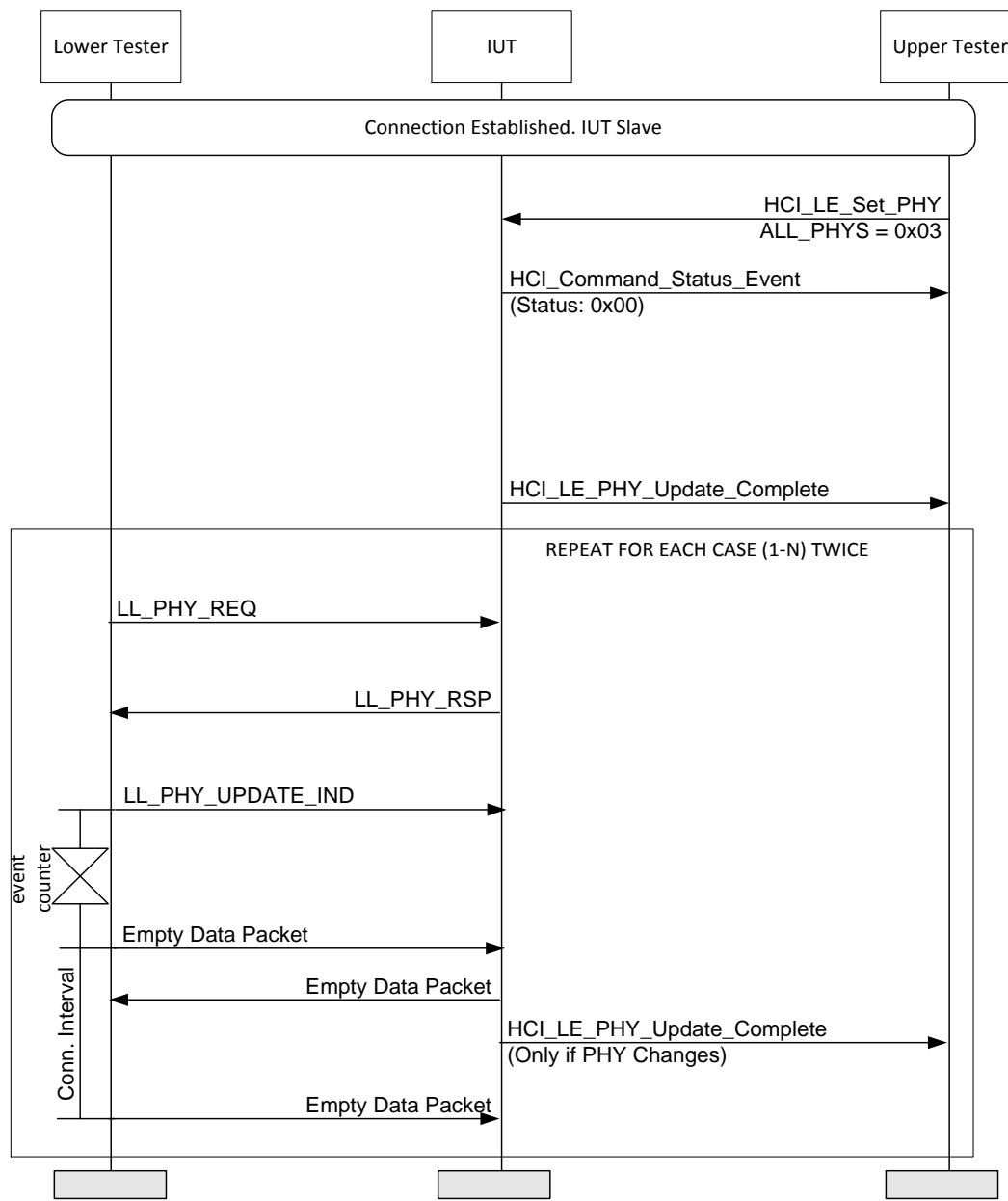


Figure 4.178: LL/CON/SLA/BV-42-C [Responding to PHY Update Procedure]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with the **ALL_PHYS** fields set to a value of **0x03**. Upper Tester expects an **HCI_Command_Status** event indicating success in response.
- The Upper Tester expects an **HCI_LE_PHY_Update_Complete** event from the IUT.
- Perform steps 4 through 11 $2N$ times as follows, where N is the number of cases in [Table 4.34](#), [Table 4.35](#), or [Table 4.36](#) (selected based on the supported PHY(s)):
 - firstly using cases 1 to N from the relevant table in order;
 - then using the cases from the relevant table in a random order.
- Lower Tester sends an **LL_PHY_REQ** PDU to the IUT to initiate a PHY change with the payload defined in the **LL_PHY_REQ** section of the relevant table.

5. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS).
6. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the payload defined by the following rules:
 - a. M_TO_S_PHY = (LL_PHY_REQ TX_PHYS) & (LL_PHY_RSP RX_PHYS Field). If this value has 0 or 1 bit set, use the value. If more than one bit is set, the Lower Tester shall select the bit listed in the M_TO_S_PHY_LTPREF column of the Lower Tester preference section of the relevant table; if this table lists more than one bit, it shall select the first bit (in the order given) that is set.
 - b. S_TO_M_PHY = (LL_PHY_REQ RX_PHYS) & (LL_PHY_RSP TX_PHYS Field). If this value has 0 or 1 bit set, use the value. If more than one bit is set, the Lower Tester shall select the bit listed in the S_TO_M_PHY_LTPREF column of the Lower Tester preference section of the relevant table; if this table lists more than one bit, it shall select the first bit (in the order given) that is set.
 - c. If IUT specifies the same single PHY in both the RX_PHYS and TX_PHYS fields, the Lower Tester shall use the PHY selected by the IUT for both directions (if allowed by the LL_PHY_REQ) or make no change. This rule shall take precedence over the values determined in rules (a) and (b).
 - d. If any of rules (a) to (c) result in a PHY equal to the current PHY, the corresponding field shall be set to zero rather than the bit corresponding to that PHY.
 - e. If either M_TO_S_PHY or S_TO_M_PHY are non-zero, then (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If M_TO_PHY and S_TO_M_PHY are both zero, the Instant shall be zero.
7. Lower Tester expects a packet from the IUT acknowledging the LL_PHY_UPDATE_IND. If both the M_TO_S_PHY and S_TO_M_PHY fields of the LL_PHY_UPDATE_IND are zero, skip to step 11.
8. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated Instant of the PHY change.
9. At the Instant of the PHY change the IUT starts maintaining the connection with the new PHY(s) selected by the Lower Tester.
10. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements. If the PHY(s) have changed, the Lower Tester shall use the new PHY(s).
11. If the PHY(s) were changed, Upper Tester expects a LE_PHY_Update_Complete event from the IUT containing the PHYs selected. If both PHYs were NOT changed, Upper Tester expects NOT to receive a LE_PHY_Update_Complete event.

Case	LL_PHY_REQ		Lower Tester preferences	
	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
1	0x02	0x02	0x02	0x02
2	0x01	0x02	0x01	0x02
3	0x02	0x01	0x02	0x01
4	0x01	0x01	0x01	0x01
5	0x03	0x02	0x01	0x02
6	0x03	0x01	0x02	0x01
7	0x01	0x03	0x01	0x02



Case	LL_PHY_REQ		Lower Tester preferences	
	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
8	0x02	0x03	0x02	0x01
9	0x03	0x03	0x02	0x02

Table 4.34: PDU payload contents for each case variation for LE 2M PHY supported and LE Coded PHY not supported.

Case	LL_PHY_REQ		Lower Tester preferences	
	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
1	0x04	0x04	0x04	0x04
2	0x01	0x04	0x01	0x04
3	0x04	0x01	0x04	0x01
4	0x01	0x01	0x01	0x01
5	0x05	0x04	0x01	0x04
6	0x05	0x01	0x04	0x01
7	0x01	0x05	0x01	0x04
8	0x04	0x05	0x04	0x01
9	0x05	0x05	0x04	0x04

Table 4.35: PDU payload contents for each case variation for LE Coded PHY supported and LE 2M PHY not supported.

Case	LL_PHY_REQ		Lower Tester preferences	
	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
1	0x03	0x01	0x02	0x01
2	0x05	0x02	0x01	0x02
3	0x02	0x04	0x02	0x04
4	0x01	0x02	0x01	0x02
5	0x04	0x01	0x04	0x01
6	0x03	0x06	0x02	0x02
7	0x01	0x01	0x01	0x01
8	0x04	0x03	0x04	0x01
9	0x05	0x01	0x01	0x01
10	0x04	0x04	0x04	0x04
11	0x05	0x07	0x04	0x02, 0x04
12	0x05	0x05	0x04	0x01
13	0x04	0x02	0x04	0x02
14	0x03	0x07	0x01	0x04, 0x01
15	0x06	0x06	0x02	0x04
16	0x03	0x02	0x02	0x02



Case	LL_PHY_REQ		Lower Tester preferences	
	TX_PHYS	RX_PHYS	M_TO_S_PHY_LTPREF	S_TO_M_PHY_LTPREF
17	0x01	0x06	0x01	0x04
18	0x05	0x06	0x01	0x04
19	0x04	0x05	0x04	0x01
20	0x01	0x05	0x01	0x04
21	0x05	0x03	0x01	0x02
22	0x01	0x04	0x01	0x04
23	0x01	0x03	0x01	0x02
24	0x03	0x05	0x02	0x01
25	0x06	0x04	0x02	0x04
26	0x02	0x07	0x02	0x01, 0x02
27	0x06	0x01	0x02	0x01
28	0x02	0x02	0x02	0x02
29	0x03	0x04	0x01	0x04
30	0x07	0x03	0x04, 0x01	0x01
31	0x02	0x01	0x02	0x01
32	0x03	0x03	0x01	0x01
33	0x02	0x03	0x02	0x01
34	0x04	0x07	0x04	0x02, 0x04
35	0x07	0x04	0x01, 0x04	0x04
36	0x07	0x01	0x04, 0x02	0x01
37	0x06	0x05	0x04	0x01
38	0x02	0x06	0x02	0x04
39	0x07	0x07	0x01, 0x02	0x01, 0x04
40	0x04	0x06	0x04	0x02
41	0x02	0x05	0x02	0x01
42	0x06	0x02	0x04	0x02
43	0x07	0x02	0x01, 0x02	0x02
44	0x07	0x06	0x04, 0x01	0x04
45	0x06	0x07	0x02	0x02, 0x01
46	0x06	0x03	0x02	0x02
47	0x05	0x04	0x04	0x04
48	0x07	0x05	0x04, 0x02	0x01
49	0x01	0x07	0x01	0x04, 0x02

Table 4.36: PDU payload contents for each case variation for both LE 2M PHY and LE Coded PHY supported.



- Expected Outcome

Pass Verdict

For all cases described in the test procedure, the following conditions shall occur:

- The IUT responds to the Lower Tester's LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHYS, RX_PHYS).
- The test procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and adopting the new PHY(s) selected by the Lower Tester at the assigned event if the PHY(s) are changed, or using the prior PHY(s) if the PHY(s) are NOT changed.
- If the PHY(s) are changed, the IUT reports the selected PHY with a LE_PHY_Update_Complete event. If both PHYs are NOT changed, the IUT does NOT send a LE_PHY_Update_Complete event. If a LE_PHY_Update_Complete event is sent, its fields are consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in LL_PHY_UPDATE_IND was zero).

Inconclusive Verdict

The PHY does not change (equivalently, step 9 is not carried out) at least once during this test case because of the rules in step 6.

4.3.4.32 LL/CON/SLA/BV-43-C [Responding to PHY Update Procedure – Symmetric Only]

- Test Purpose

Test that a slave IUT is able to respond to a PHY update procedure from a master device when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the master role maintaining a connection and initiates the PHY update procedure. IUT responds to the PHY change request and notifies the host only when a change occurs.

- Reference

[10] 5.1.10

- Initial Condition

Same as [LL/CON/SLA/BV-42-C \[Responding to PHY Update Procedure\]](#).

- Test Procedure

Same as [LL/CON/SLA/BV-42-C \[Responding to PHY Update Procedure\]](#), except that the Lower Tester always selects a single (symmetric) PHY for both directions in the LL_PHY_UPDATE_IND PDU.

- Expected Outcome

Pass Verdict

Same as [LL/CON/SLA/BV-42-C \[Responding to PHY Update Procedure\]](#) except that the IUT shall only set a single (symmetric) PHY in each LL_PHY_RSP PDU.



Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case because of the rules in step 4 of the test procedure in [LL/CON/SLA/BV-42-C \[Responding to PHY Update Procedure\]](#).

4.3.4.33 LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT's PHY change request and the Lower Tester's PHY change request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY update procedure upon receiving the IUT's PHY change to cause a procedure collision. The test case expects the IUT to respond to the master's PHY update procedure after the master has rejected the IUT's PHY update procedure.

- Reference

[\[10\]](#) 5.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to `LL_slave_connSlaveLatency_MAX`, selected timeout, any SCA value).



- Test Procedure

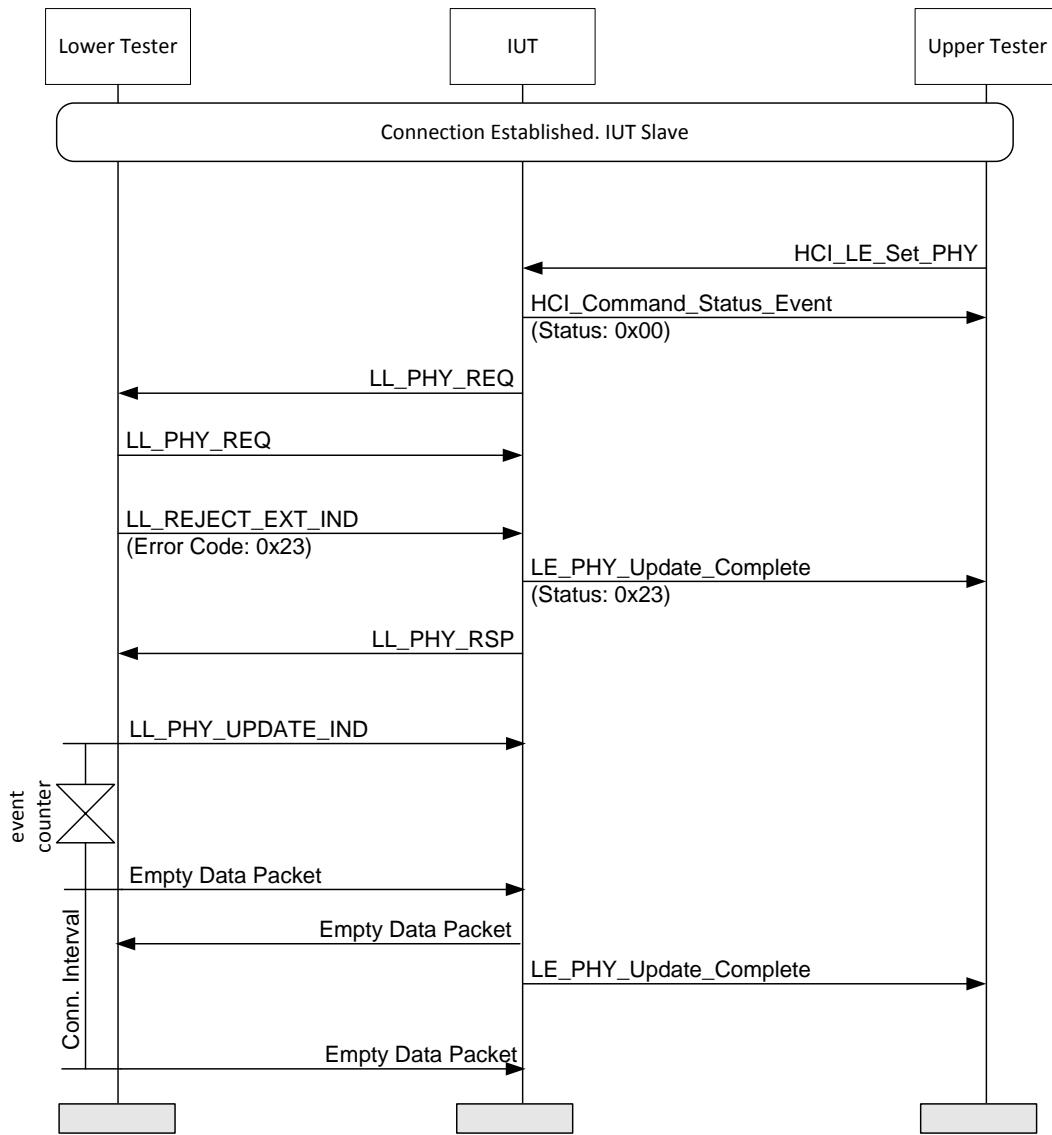


Figure 4.179: LL/CON/SLA/BV-44-C [Handling Protocol Collision – Same Procedure]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer PHY other than LE 1M, and PHY_options set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
- Lower Tester expects a **LL_PHY_REQ** PDU from the IUT. Lower Tester responds with a **LL_PHY_REQ** with TX_PHYS and RX_PHYS set to prefer a PHY other than LE 1M and rejects the IUT's **LL_PHY_REQ** PDU using a **LL_REJECT_EXT_IND** with ErrorCode 0x23 (LMP Error Transaction Collision). If the IUT does not send an **LL_PHY_REQ** PDU the test case ends with an Inconclusive Verdict.
- The IUT may send an **HCI_LE_PHY_Update_Complete** event with status set to 0x23 to the Upper Tester.
- The IUT responds to the **LL_PHY_REQ** from the Lower Tester with a **LL_PHY_RSP** with at least one bit set in each field (RX_PHYS, TX_PHYS).
- The Lower Tester responds with an **LL_PHY_UPDATE_IND** PDU with M_TO_S_PHY and S_TO_M_PHY fields set to a PHY other than LE 1M and with an Instant field set such that

$(\text{Instant} - \text{connEventCount}) \bmod 65536$ shall be less than 32767 and greater than 6. The Lower Tester expects a packet from the IUT acknowledging the LL_PHY_UPDATE_IND.

6. The Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
7. At the Instant of the PHY change start maintaining the connection with the selected PHYs.
8. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements. If PHY has changed, the Lower Tester shall use the new PHYs.
9. The Upper Tester expects an HCI_LE_PHY_Update_Complete from the IUT indicating that both RX_PHY and TX_PHY match the settings used in step 2.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_PHY_REQ PDU to update the selected PHYs.

If the IUT reports an HCI_LE_PHY_Update_Complete event when the Lower Tester rejects the IUT's request, it shall have the correct error code (0x23).

The IUT responds to the LL_PHY_REQ from the Lower Tester with a LL_PHY_RSP after receiving the LL_REJECT_EXT_IND.

The test procedure is executed successfully, with the IUT acknowledging the LL_PHY_UPDATE_IND and adopting the new rate at the assigned event.

The IUT reports the new PHY(s) to the host with an HCI_LE_PHY_Update_Complete event.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.34 LL/CON/SLA/BV-45-C [Protocol Timeout for PHY Update Procedure]

- Test Purpose

Test that a slave IUT terminates the Link Layer connection if the slave-initiated PHY update procedure is not completed before the procedure response timer expires.

The Lower Tester acts in the master role in the connection and ensures that the procedure initiated by the IUT is not completed.

- Reference

[10] 5.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

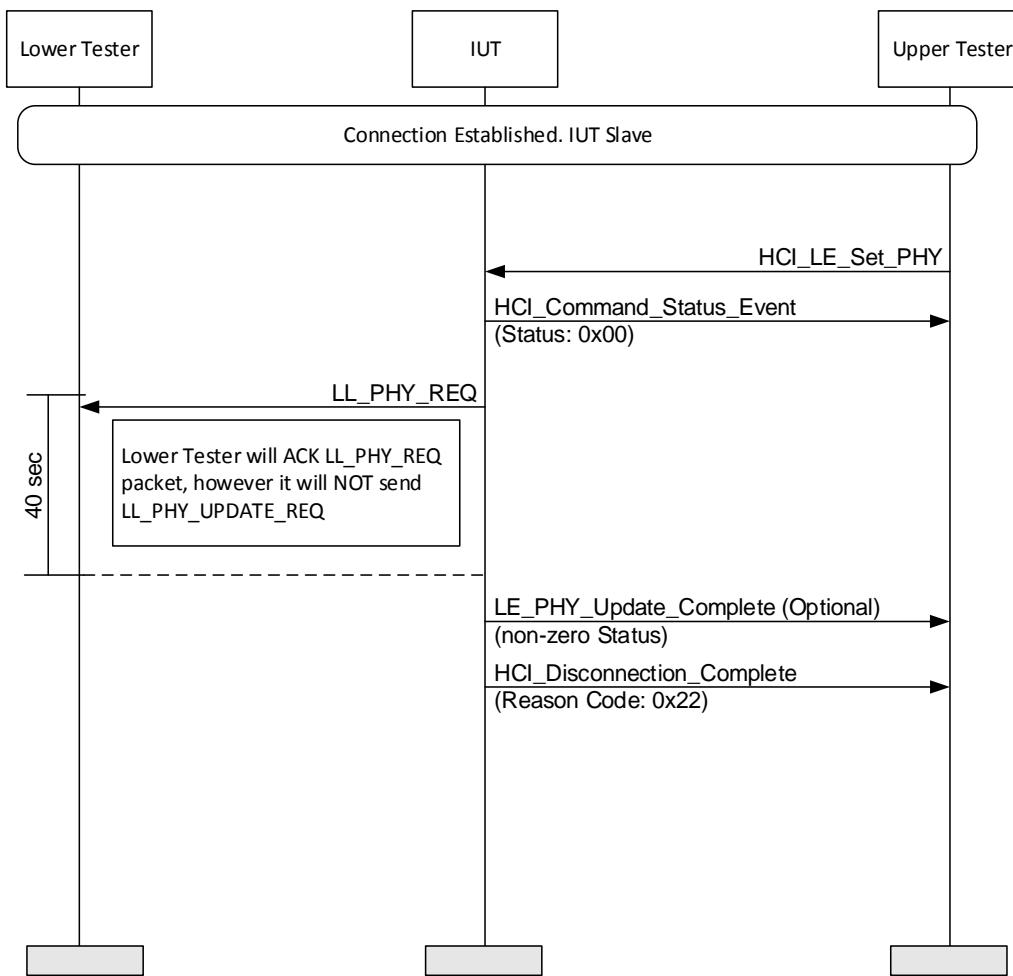


Figure 4.180: LL/CON/SLA/BV-45-C [Protocol Timeout for PHY Update Procedure]

1. Upper Tester sends an HCI_LE_Set_PHY with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M and PHY_options set to 0x0000 to the IUT and receives an HCI_Command_Status_Event.
 2. Lower Tester expects a LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
 3. Lower Tester acknowledges the LL_PHY_REQ PDU but does not send an LL_PHY_UPDATE_IND PDU.
 4. The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero Status.
 5. IUT sends the HCI_Disconnection_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.
- Expected Outcome

Pass Verdict

The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in Step 4 the status shall be non-zero.



Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.35 LL/CON/SLA/BV-46-C [Handling Protocol Collision – Different Procedure – Channel Map]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT's PHY change request and the remote device's channel map update.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates channel map update procedure upon receiving the IUT's PHY change to cause a procedure collision. The test case expects the IUT to respond to the master's channel map update procedure after the master has rejected the IUT's PHY update procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

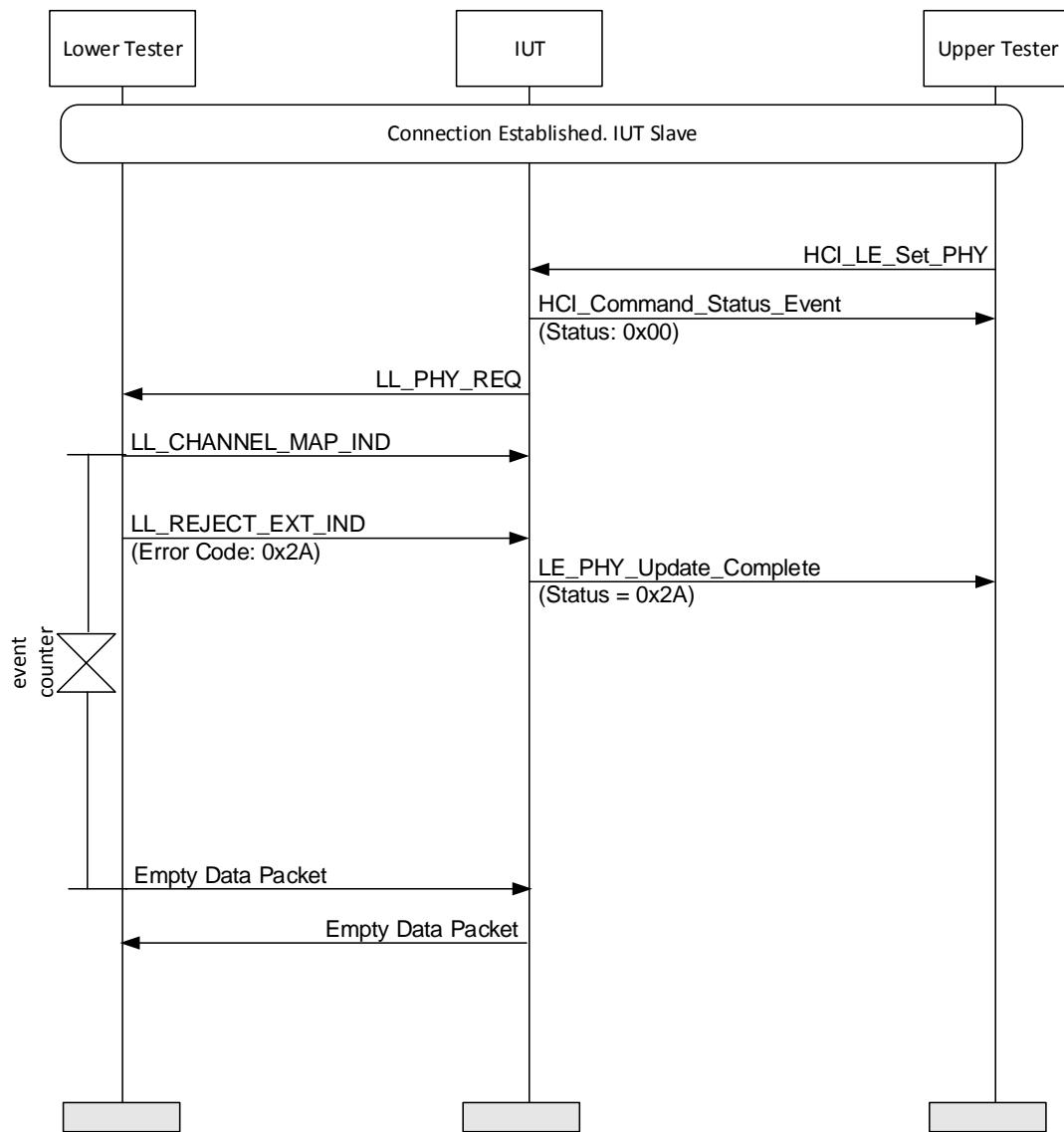


Figure 4.181: LL/CON/SLA/BV-46-C [Handling Protocol Collision – Different Procedure – Channel Map]

- Upper Tester sends an HCl LE Set PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and expects an HCl_Command_Status event from the IUT in response.
- Lower Tester expects a LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
- Lower Tester responds with an LL_CHANNEL_MAP_IND and rejects the IUT's LL_PHY_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
- The IUT sends the HCl LE PHY Update Complete event with status set to 0x2A to the Upper Tester.
- Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of channel map update.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_PHY_REQ PDU to update the PHY and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT's request. The IUT properly handles the master initiated channel map request procedure.

The test procedure is executed successfully, with the IUT acknowledging the channel map request and adopting the new channel map at the assigned event.

The IUT maintains the connection with the Lower Tester after the channel map update completes.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.36 LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT's PHY change request and the remote device's connection parameters request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameters request procedure upon receiving the IUT's PHY change to cause a procedure collision. The test case expects the IUT to respond to the master's connection parameters request procedure after the master has rejected the IUT's PHY update procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

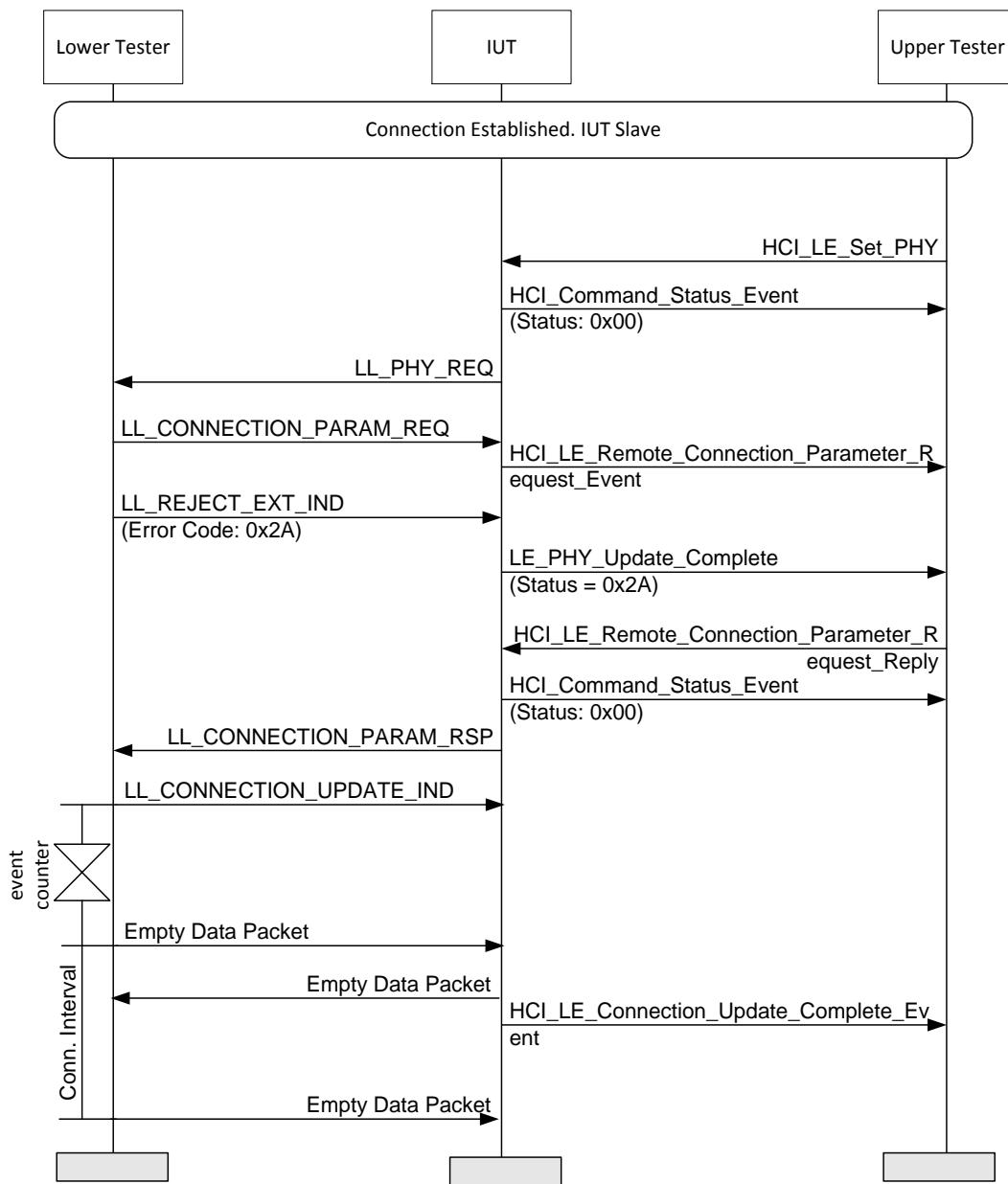


Figure 4.182: LL/CON/SLA/BV-47-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
- Lower Tester expects a **LL_PHY_REQ** PDU from the IUT. If the IUT does not send an **LL_PHY_REQ** PDU the test case ends with an Inconclusive Verdict.
- Lower Tester responds with an **LL_CONNECTION_PARAM_REQ** and rejects the IUT's **LL_PHY_REQ** PDU using an **LL_REJECT_EXT_IND** with ErrorCode 0x2A.
- The IUT requests the Upper Tester to accept or reject the Lower Tester's parameter update request.
- IUT sends the **HCI_LE_PHY_Update_Complete** event with status set to 0x2A to the Upper Tester.



6. The Upper Tester accepts the IUT's request to accept or reject the Lower Tester's parameter update request.
7. The IUT responds to the LL_CONNECTION_PARAM_REQ from the Lower Tester with an LL_CONNECTION_PARAM_RSP.
8. The Lower Tester responds with an LL_CONNECTION_UPDATE_IND PDU and expects a packet from the IUT acknowledging the connection update request.
9. Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
10. At the time of the update start maintaining the connection with the new parameters.
11. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_PHY_REQ PDU to update the selected PHYs and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT's request.

The IUT responds positively to the master initiated connection parameters request procedure.

The test procedure is executed successfully, with the IUT acknowledging the connection update and adopting the new parameters at the assigned event.

The IUT maintains the connection with the Lower Tester after the connection update completes.

The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.37 LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update]

- Test Purpose

Test that a slave IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT's PHY change request and the remote device's connection update request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester then observes the procedure carried out by the IUT and initiates a connection update procedure upon receiving the IUT's PHY change to cause a procedure collision. The test case expects the IUT to respond to the master's connection update request procedure after the master has rejected the IUT's PHY update procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

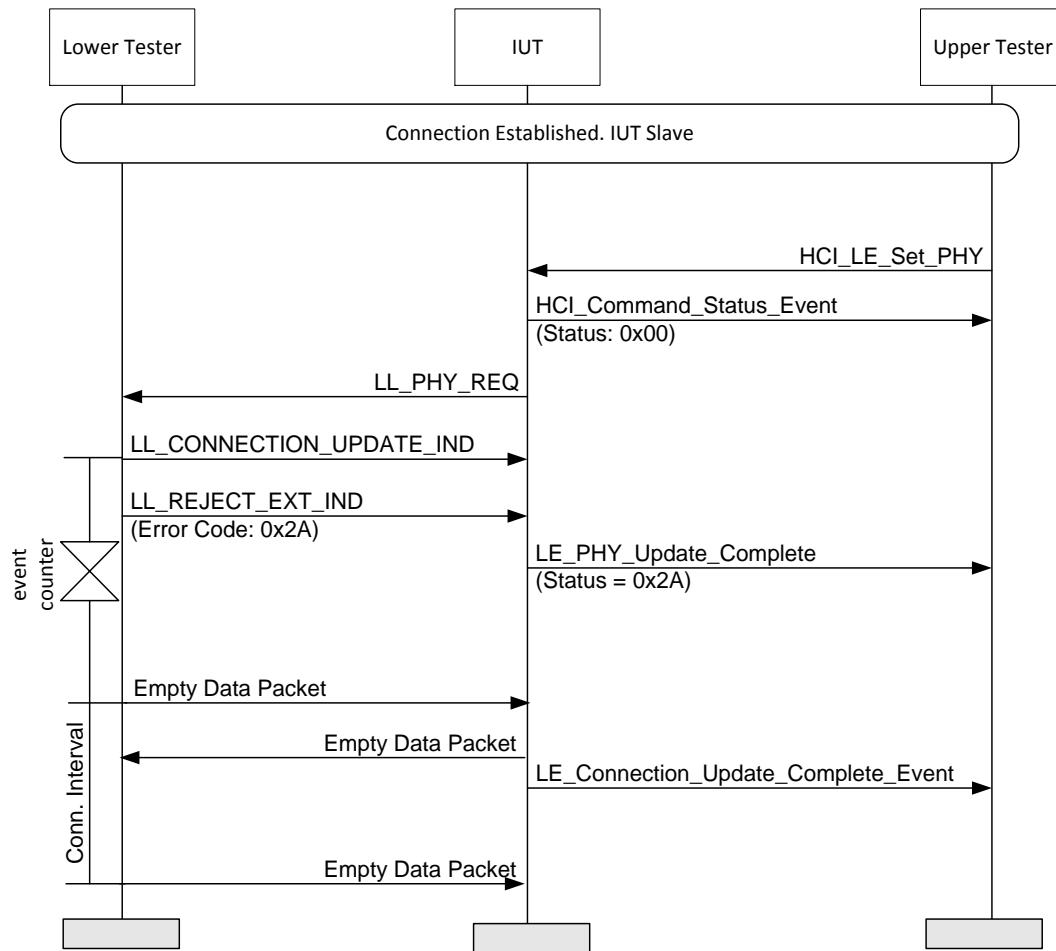


Figure 4.183: LL/CON/SLA/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Update]

- Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
- Lower Tester expects a LL_PHY_REQ PDU from the IUT. If the IUT does not send an LL_PHY_REQ PDU the test case ends with an Inconclusive Verdict.
- Lower Tester responds with an LL_CONNECTION_UPDATE_IND and rejects the IUT's LL_PHY_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
- IUT sends the HCI_LE_PHY_Update_Complete event with status set to 0x2A to the Upper Tester.
- Lower Tester sends empty DATA packets to the IUT, expecting acknowledgements until the event count matches the indicated time of connection update.
- At the time of the update start maintaining the connection with the new parameters.
- Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_PHY_REQ PDU to update the selected PHY(s) and reports an HCI_LE_PHY_Update_Complete event with the correct reason code (0x2A) when the Lower Tester rejects the IUT's request.

The IUT responds positively to the master initiated connection update request procedure.

The test procedure is executed successfully, with the IUT acknowledging the connection update and adopting the new parameters at the assigned event.

The IUT maintains the connection with the Lower Tester after the connection update completes.

The IUT reports the new connection parameters with an HCI_LE_Connection_Update_Complete event.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.4.38 LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it initiates the PHY update procedure.

The Lower Tester, in the master role, maintains the connection.

A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY update procedure as the Host of the IUT.

- Reference

[10] 5.1.10.1

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

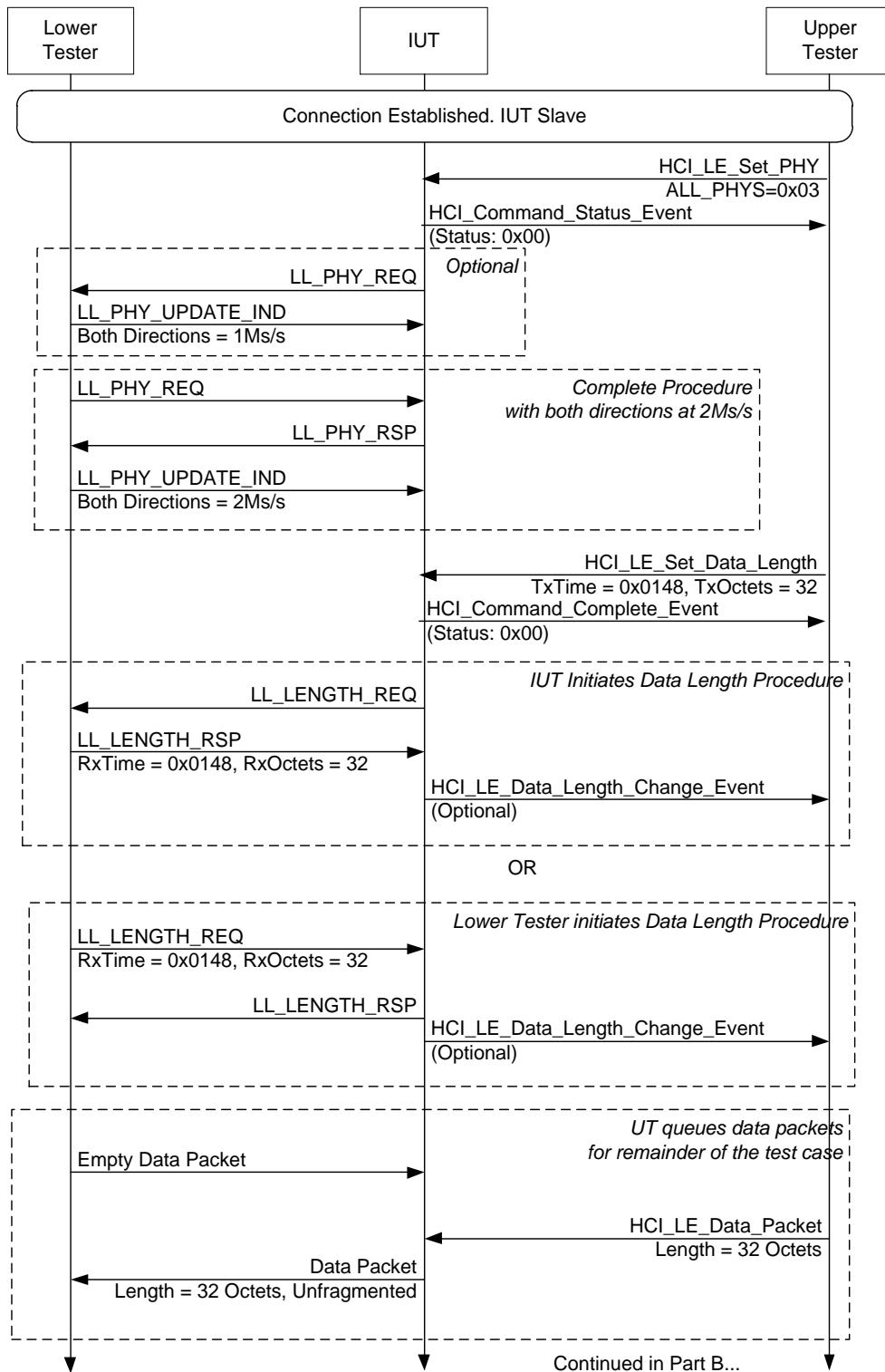


Figure 4.184: LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions – Part A]

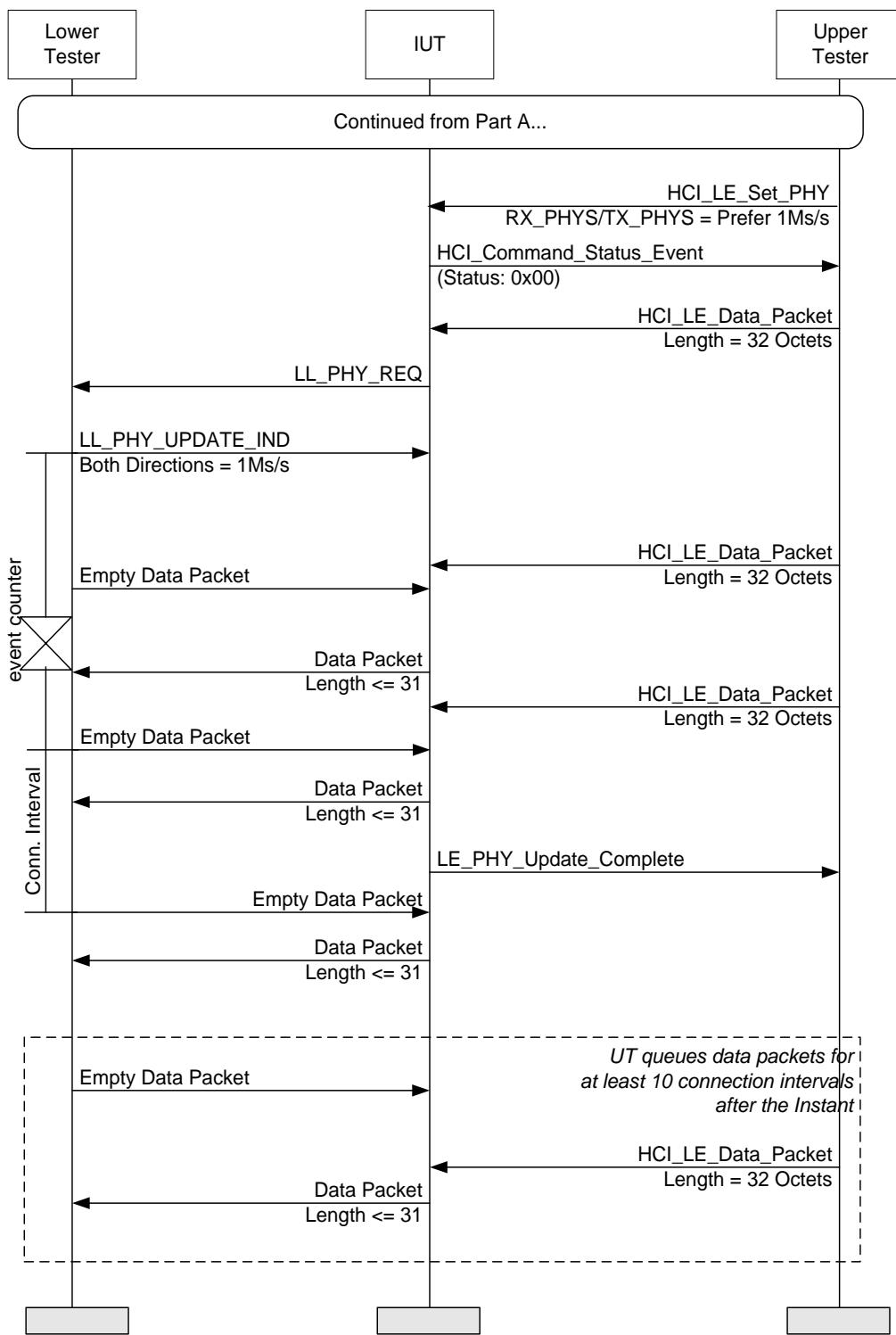


Figure 4.185: LL/CON/SLA/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions - Part B]

1. Upper Tester sends an **HCI LE_Set_PHY** command to the IUT with **ALL_PHYS** set to **0x03**, and expects an **HCI_Command_Status** event from the IUT in response.
2. The Upper Tester expects an **LE_PHY_Update_Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.

3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
 4. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
 5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
 6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and expects an HCI_Command_Complete event from the IUT in response.
 7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
 8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 9. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 10. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 1M PHY.
 11. The Lower Tester expects an LL_PHY_REQ from the IUT which allows for selection of the LE 1M PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.
 12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 11) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY update procedure.
 13. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 14. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.
 15. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
 16. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in Step 11) must have a length of 32 octets and be un-fragmented.



All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in Step 11) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not allow the Lower Tester to select the LE 2M PHY in Step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in Step 11).
- The IUT does not allow the Lower Tester to select the LE 1M PHY in Step 11).

4.3.4.39 LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions]

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY update procedure from a master.

The Lower Tester, in the master role, maintains the connection.

A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY update procedure.

- Reference
- [10] 5.1.10.1
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

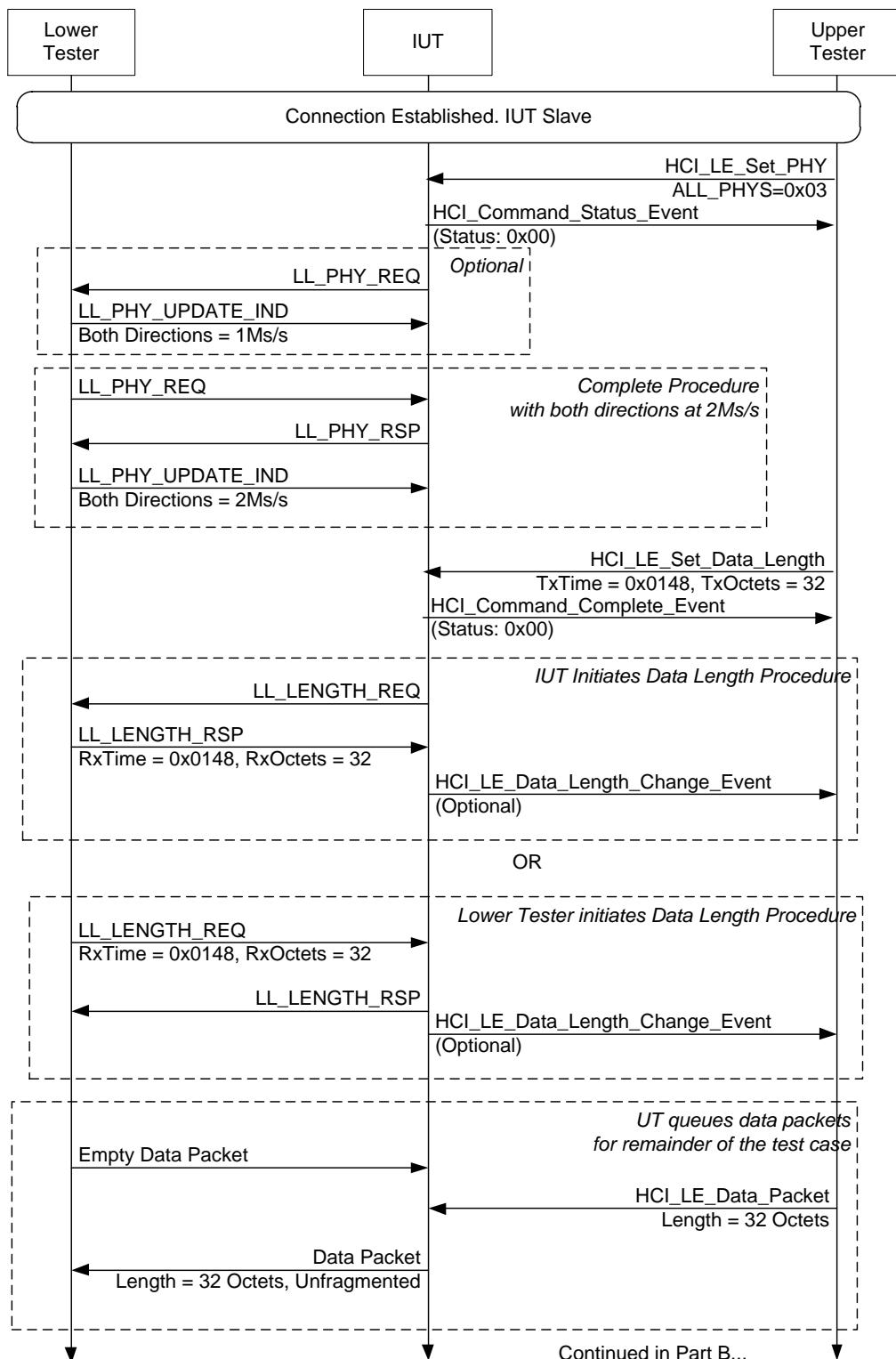


Figure 4.186: LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions – Part A]

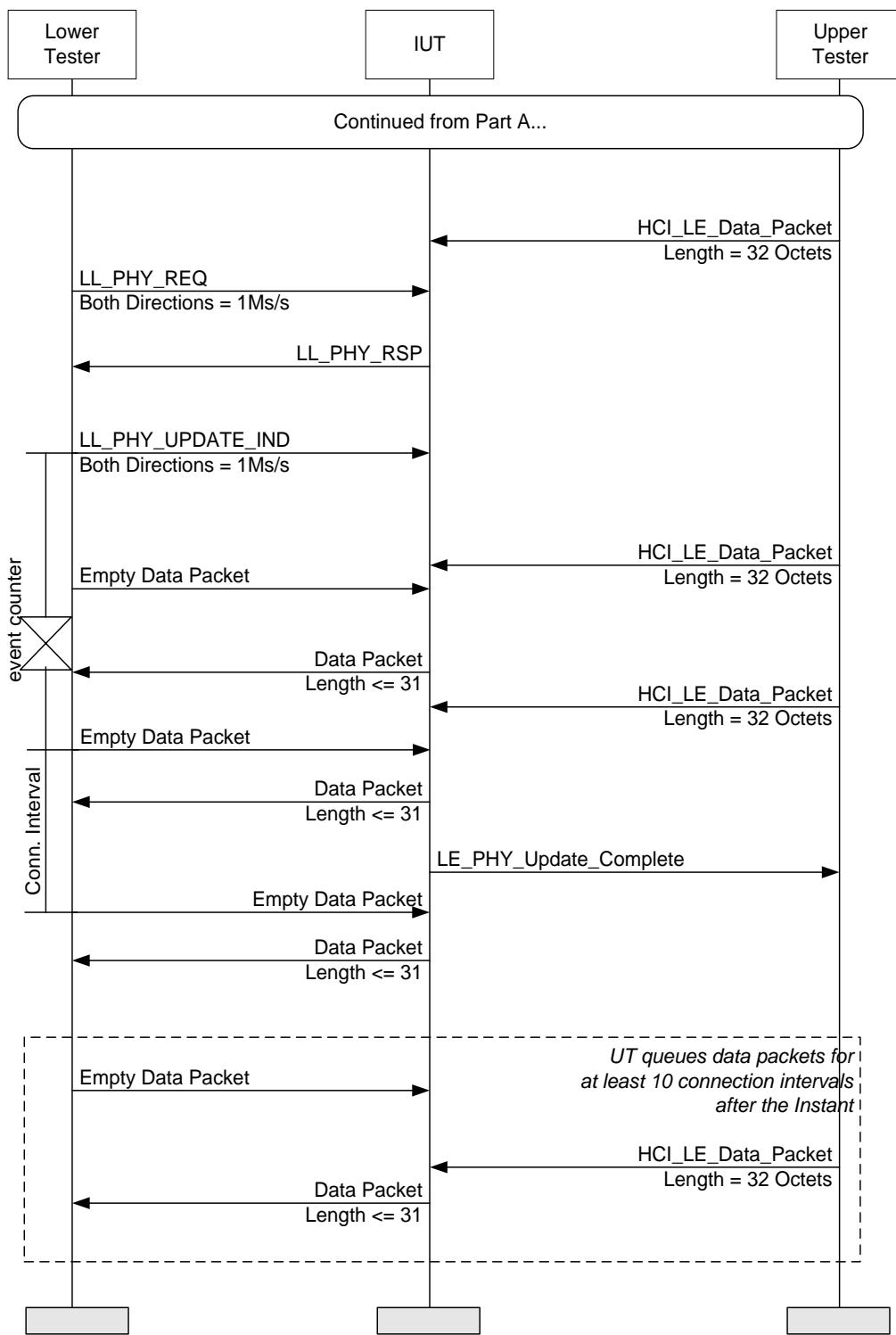


Figure 4.187: LL/CON/SLA/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions - Part B]

1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with **ALL_PHYS** set to **0x03**, and expects an **HCI_Command_Status** event from the IUT in response.
2. The Upper Tester expects an **HCI_LE_PHY_Update_Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.

3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
 4. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
 5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
 6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and expects an HCI_Command_Complete event from the IUT in response.
 7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
 8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 9. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 10. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x01.
 11. The Lower Tester expects an LL_PHY_RSP from the IUT which allows for selection of the LE 1M PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.
 12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 11) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY update procedure.
 13. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 14. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.
 15. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
 16. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in Step 11) must have a length of 32 octets and be un-fragmented.



All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in Step 11) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not allow the Lower Tester to select the LE 2M PHY in Step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in Step 11).
- The IUT does not allow the Lower Tester to select the LE 1M PHY in Step 11).

4.3.4.40 LL/CON/SLA/BV-51-C [Protocol Timeout for PHY Update Procedure – No Update Request]

- Test Purpose

Test that a slave IUT terminates the Link Layer connection if master-initiated PHY update procedure is not completed before the procedure response timer expires.

The Lower Tester acts in the master role in the connection and ensures that the procedure initiated by the Lower Tester is not completed.

- Reference

[10] 5.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

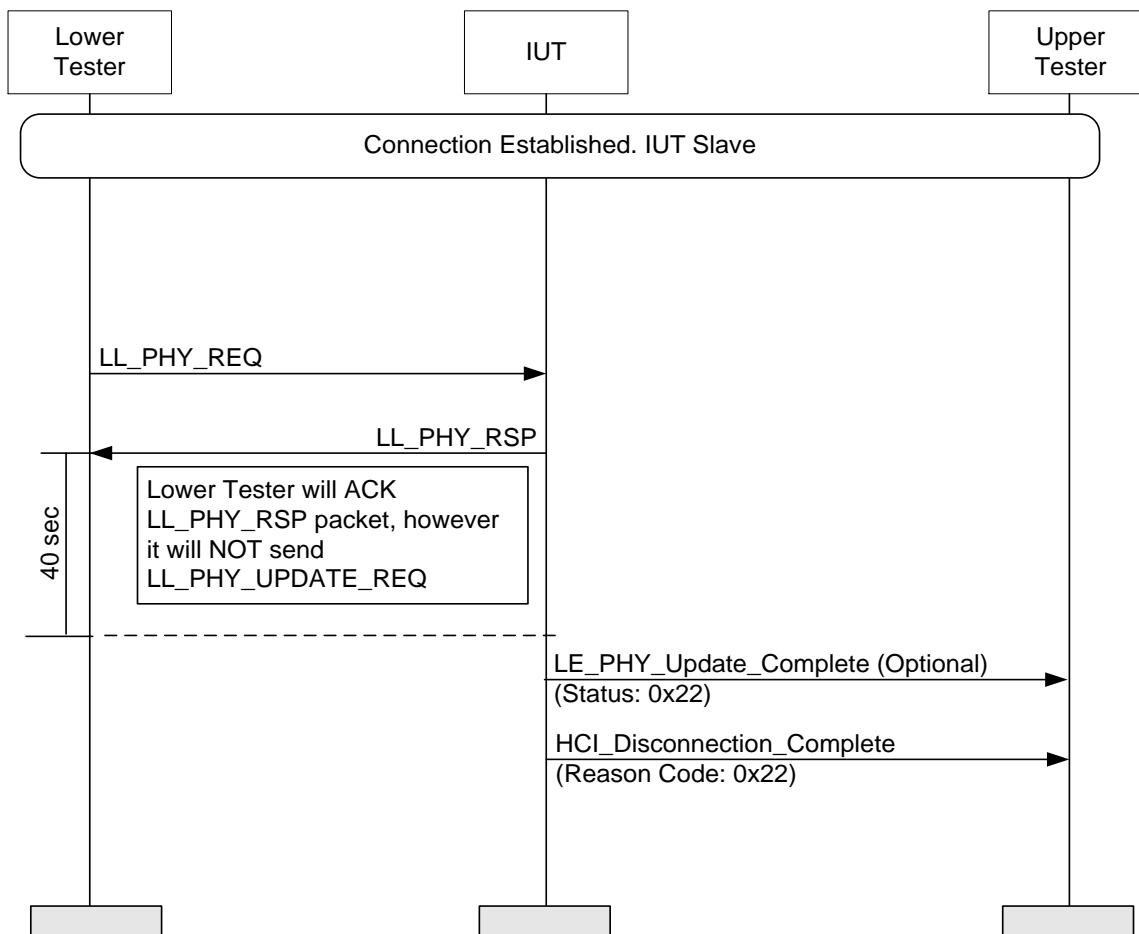


Figure 4.188: LL/CON/SLA/BV-51-C [Protocol Timeout for PHY Update Procedure]

- The Lower Tester sends an LL_PHY_REQ PDU with TX_PHYS and RX_PHYS both set to prefer a PHY other than LE 1M.
 - Lower Tester expects an LL_PHY_RSP from the IUT, and acknowledges this packet, but does not send an LL_PHY_UPDATE_IND PDU.
 - The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero status.
 - IUT sends the HCI_Disconnection_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.
- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHYS, RX_PHYS).

The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in Step 3 the status shall be non-zero.



4.3.4.41 LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change]

- **Test Purpose**

Tests that a slave IUT follows all packet time restrictions when a PHY update procedure is initiated but no PHY change occurs.

The Lower Tester, in the master role, maintains the connection.

A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY update procedure as the Host of the IUT, but the Lower Tester does not allow it to result in a PHY change.

- **Reference**

[10] 5.1.10.1

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

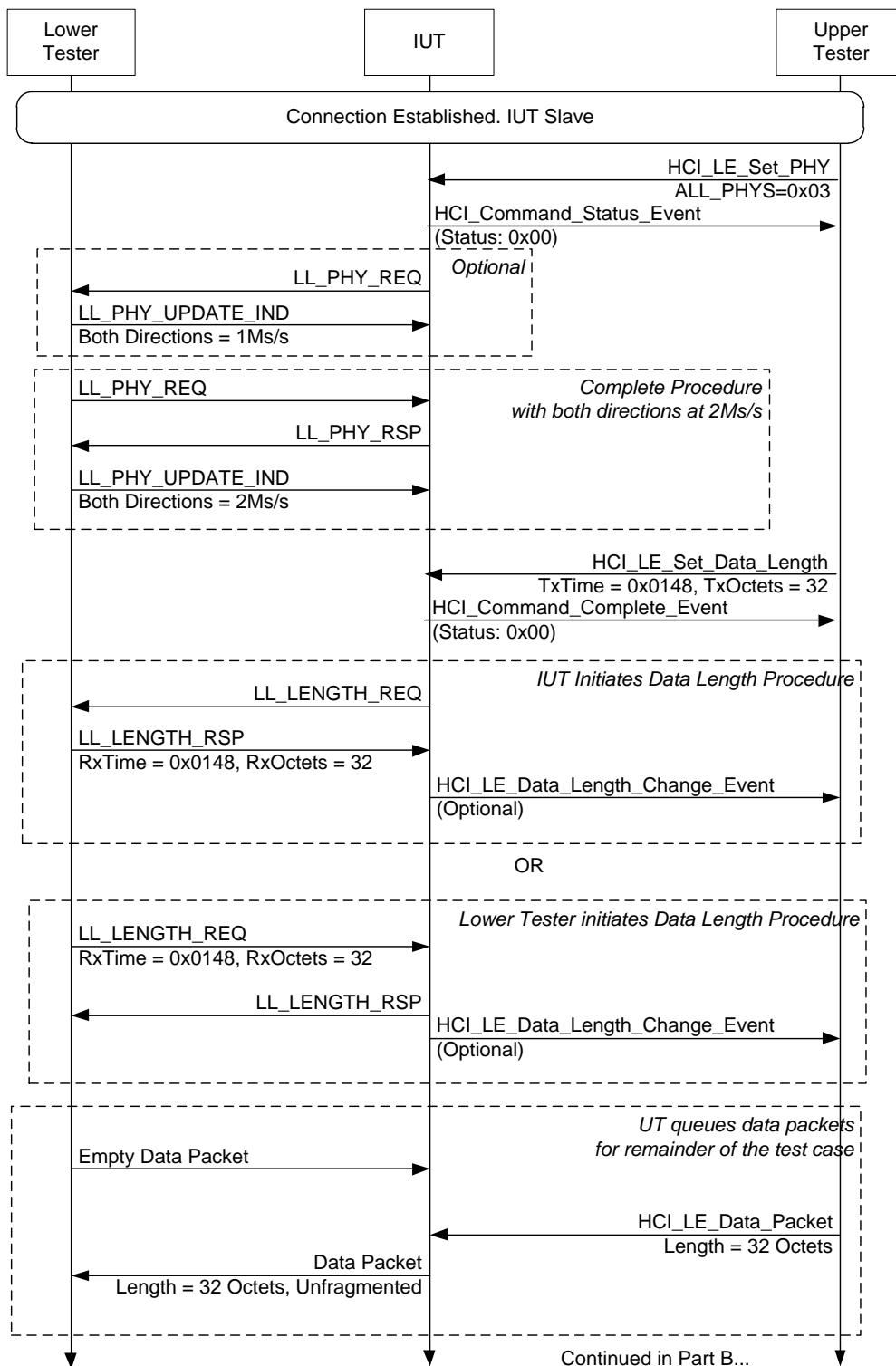


Figure 4.189: LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change – Part A]



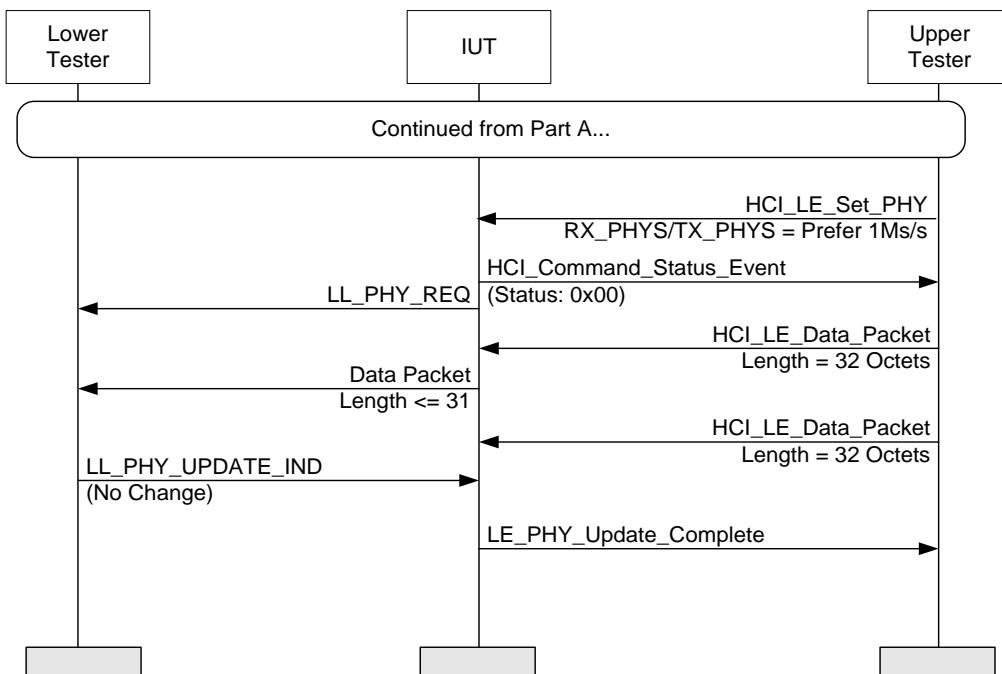


Figure 4.190: LL/CON/SLA/BV-52-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change – Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and expects an HCI_Command_Status event from the IUT in response.
 2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
 3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
 4. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
 5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
 6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32.
 7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
 8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 9. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 10. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 1M PHY.

11. The Lower Tester expects an LL_PHY_REQ from the IUT in which the TX_PHYS field has the 0x01 bit set. If the TX_PHYS field does not have the 0x01 bit set, the test ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 140 octets of data are received. If the IUT does not send 140 octets of data before the procedure timeout expire, the test ends with an Inconclusive Verdict.
 12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 11) must be 31 octets or less in length. If a larger packet is received during this period, the test case ends with a failed verdict.
 13. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).
 14. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 2M PHY.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in Step 11) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in Step 11) must have a length of less than or equal to 31 octets, until the Lower Tester sends the LL_PHY_UPDATE_IND in Step 13).

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not allow the Lower Tester to select the LE 2M PHY in Step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in Step 11).
- The IUT does not set the 0x01 bit in the TX_PHYS field in Step 11).
- The IUT does not send 140 octets of data before the procedure response supervision timeout expires.

4.3.4.42 LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change]

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY update procedure from a master but no PHY change occurs.



The Lower Tester, in the master role, maintains a connection.

A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY update procedure but does not allow it to result in a PHY change.

- Reference
 - [10] 5.1.10.1
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

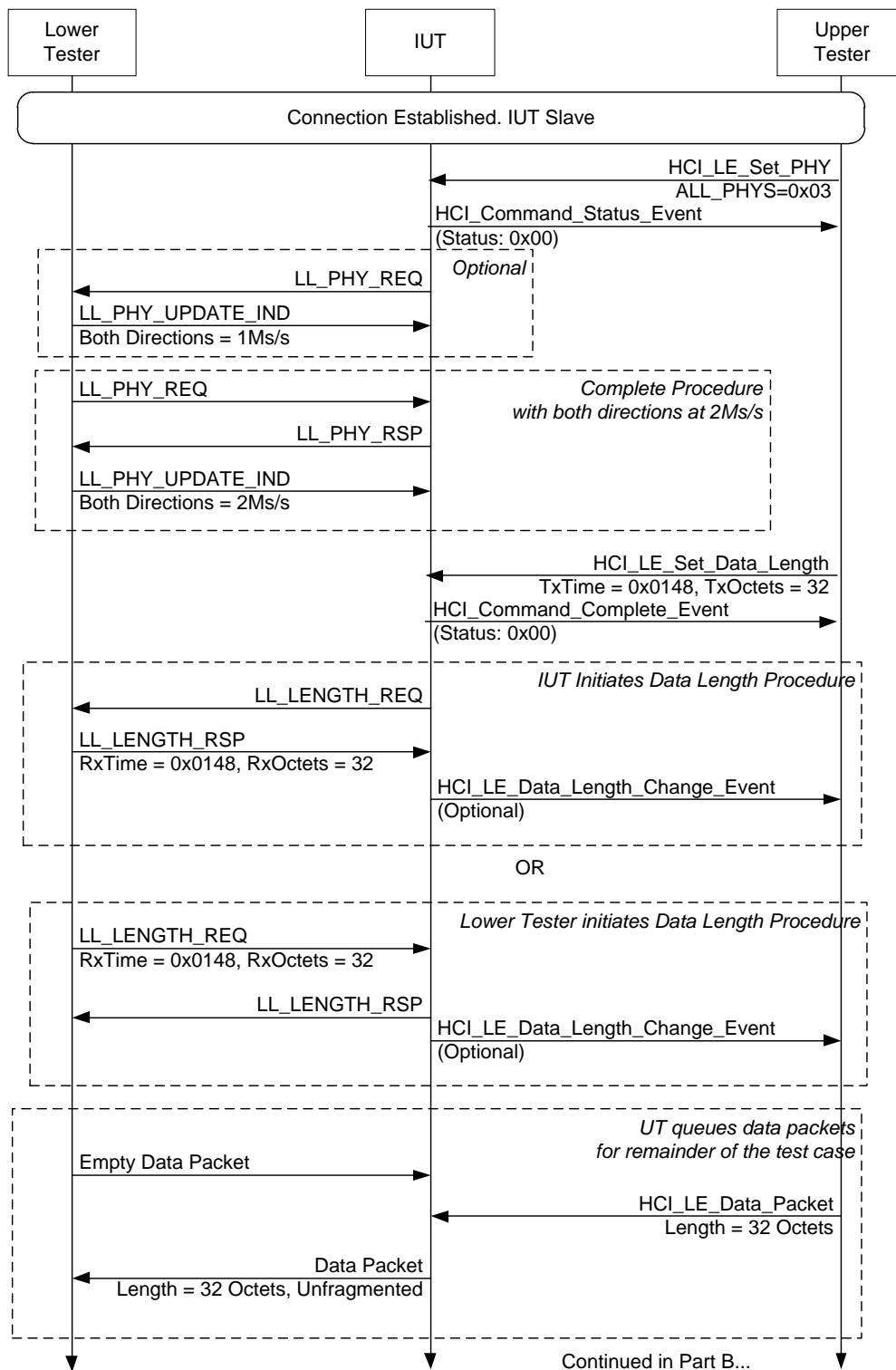


Figure 4.191: LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change – Part A]

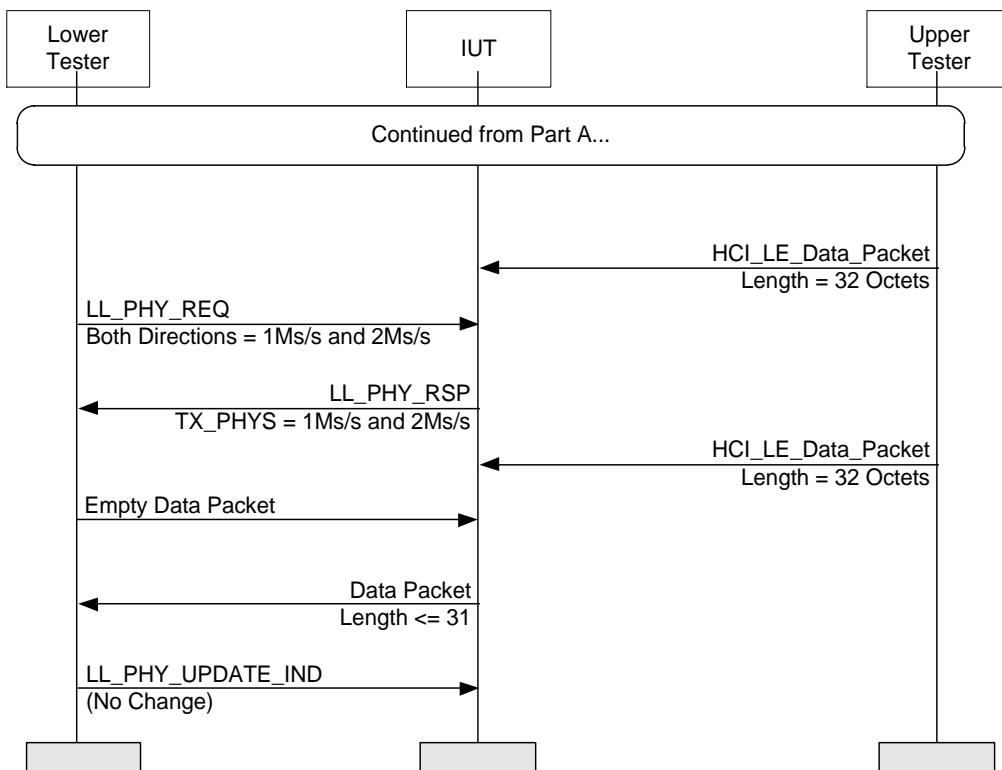


Figure 4.192: LL/CON/SLA/BV-53-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and expects an HCI_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE 2M PHY.
4. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 2M PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select the LE 2M PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
6. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and expects an HCI_Command_Complete event from the IUT in response.
7. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
8. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.

9. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 10. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x03.
 11. The Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE 1M PHY bit must be set in the TX_PHYS field by the IUT or the test case ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 140 octets are received.
If the IUT does not send 140 octets of data before the procedure timeout expire, the test ends with an Inconclusive Verdict.
 12. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 11) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict.
 13. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in Step 11) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in Step 11) must have a length of less than or equal to 31 octets until the Lower Tester sends the LL_PHY_UPDATE_IND in Step 13)

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not allow the Lower Tester to select the LE 2M PHY in Step 4).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 7).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in Step 11).
- The IUT does not set the LE 1M PHY bit in the TX_PHYS field in Step 11).
- The IUT does not send 140 octets of data before the procedure response supervision timeout expires.



4.3.4.43 LL/CON/SLA/BV-54-C [Slave Receiving Data, LE Coded, CI Change]

- **Test Purpose**

Test that a slave IUT is able to receive data from a master device when the master is transitioning between 125kbit and 500kbit coded rates. Confirm that IUT responds within the allowed T_IFS times for each packet at either coded rate. Test is performed with the IUT's minimum and maximum supported packet lengths. A Data Length Update Procedure is performed if required.

The Lower Tester acts in the master role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- **Reference**

[10] 4.5

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, maximum supported connection interval, common slave latency, common timeout, common channel map, any SCA value).

- **Test Procedure**

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



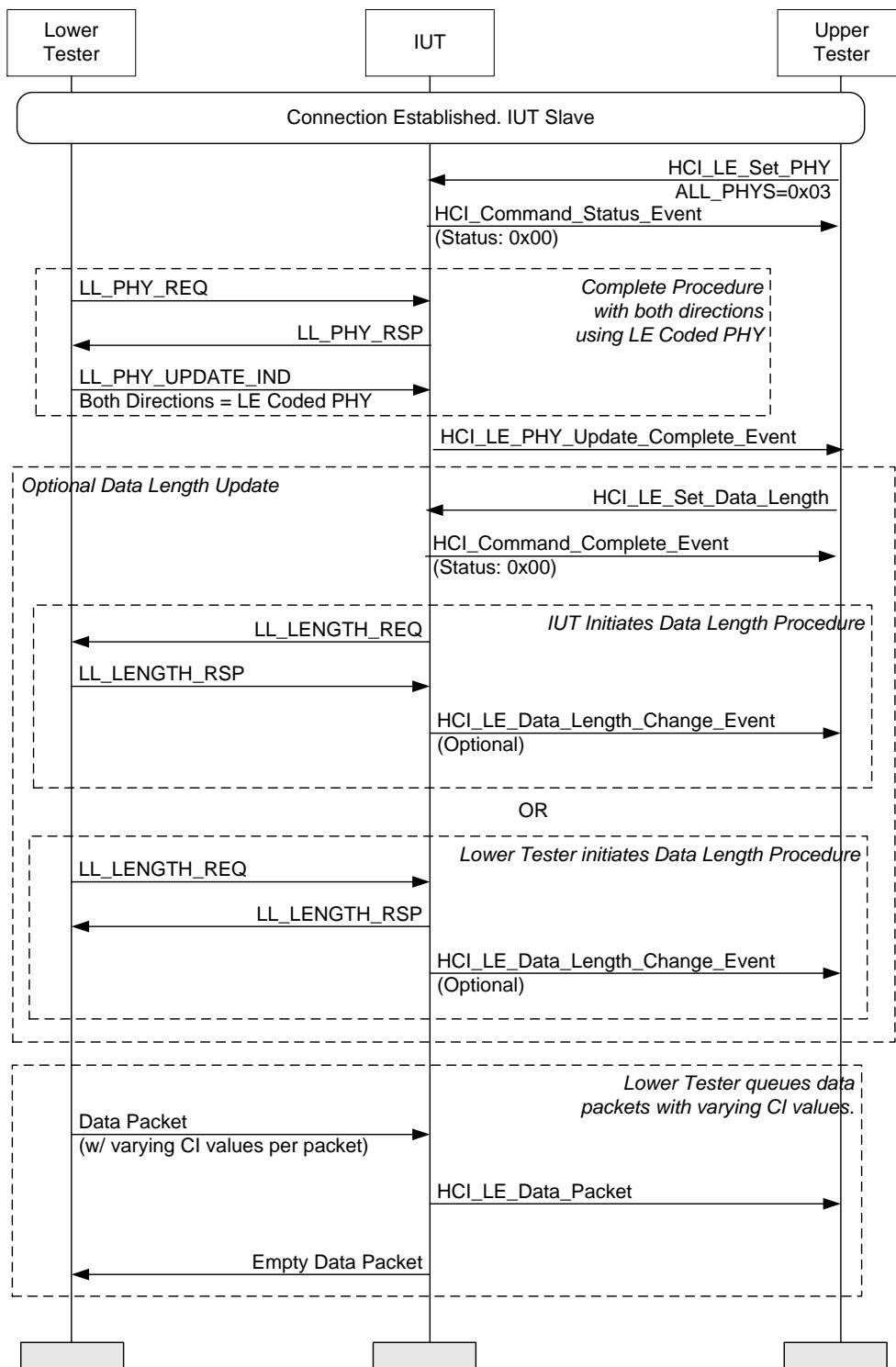


Figure 4.193: LL/CON/SLA/BV-54-C [Slave Receiving Data, LE Coded, CI Change]

1. Upper Tester sends an HCl LE Set PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCl_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCl LE Phy Update Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. Lower Tester sends an LL_PHY_REQ PDU to the IUT to initiate a PHY update with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer the LE Coded PHY.
4. Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE Coded PHY bit must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
5. Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select LE Coded PHY and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6 and completes the procedure. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE Coded PHY.
6. If (TSPX_TxOctets_Max > 27) OR (TSPX_TxTime_Max > 328) then the Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to TSPX_TxTime_Max and TxOctets set to TSPX_TxOctets_Max and expects an HCI_Command_Complete event from the IUT in response, otherwise go to step 8.
7. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max and expects a response from the IUT.
8. Configure Lower Tester to send 288 data packets of length connEffectiveMaxTxOctets with all payload octets set to continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFF, 0xFF, 0x00, 0x01, etc.). For each packet sent the CI field will alternate in the pattern below. Each number in the pattern represents the CI field for that particular packet to be sent.

```

2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 8 2 2 2 2 2 2 2 2
2 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 8 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 2 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 2 2 2 2
8 8 8 2 2 2 8 8 8 2 8 8 2 2 2 8
2 8 2 2 8 2 8 8 8 8 2 8 2 8 2
8 2 2 2 2 8 2 8 8 2 8 8 8 2 2 2
8 8 8 2 2 8 2 8 2 8 8 2 2 8 2
2 2 2 2 8 2 8 8 2 8 2 8 8 2 2 2
8 2 2 2 8 2 2 8 2 2 2 8 2 2 2 2
2 2 2 8 2 2 8 2 2 8 2 8 2 2 2 8
8 8 8 2 8 8 8 2 2 2 8 8 8 8 8

```



9. Lower Tester sends one or more DATA packets to the IUT, using the acknowledgement scheme and the data channel selection parameters. Lower Tester expects an empty DATA packet in response from the IUT for each DATA packet sent by the Lower Tester, received with acceptable TIFS timing (150 μ s +/- 2 μ s).
 10. Upper Tester expects an HCI_LE_Data_Packet event from the IUT containing one or more data elements sent in step 8. The received data should be in the form of continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFF, 0x00, 0x01, etc.).
 11. Repeat step 8 through 10 until all data to be sent has been reported to the Upper Tester.
 12. Repeat the entire data exchange procedure again from step 8 through 11, but utilizing the minimum data payload size (27 bytes) instead of connEffectiveMaxTxOctets for each data packet.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent.

The IUT reports all data correctly with HCI_LE_Data_Packet events containing continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFF, 0x00, 0x01, etc.).

All responses from the IUT are received with acceptable TIFS timing (150 μ s +/- 2 μ s).

4.3.4.44 LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it initiates the PHY Update Procedure. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester, in the master role, maintains a connection.

A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY Update Procedure as the Host of the IUT.

- Reference
[\[10\] 5.1.10.1](#)
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

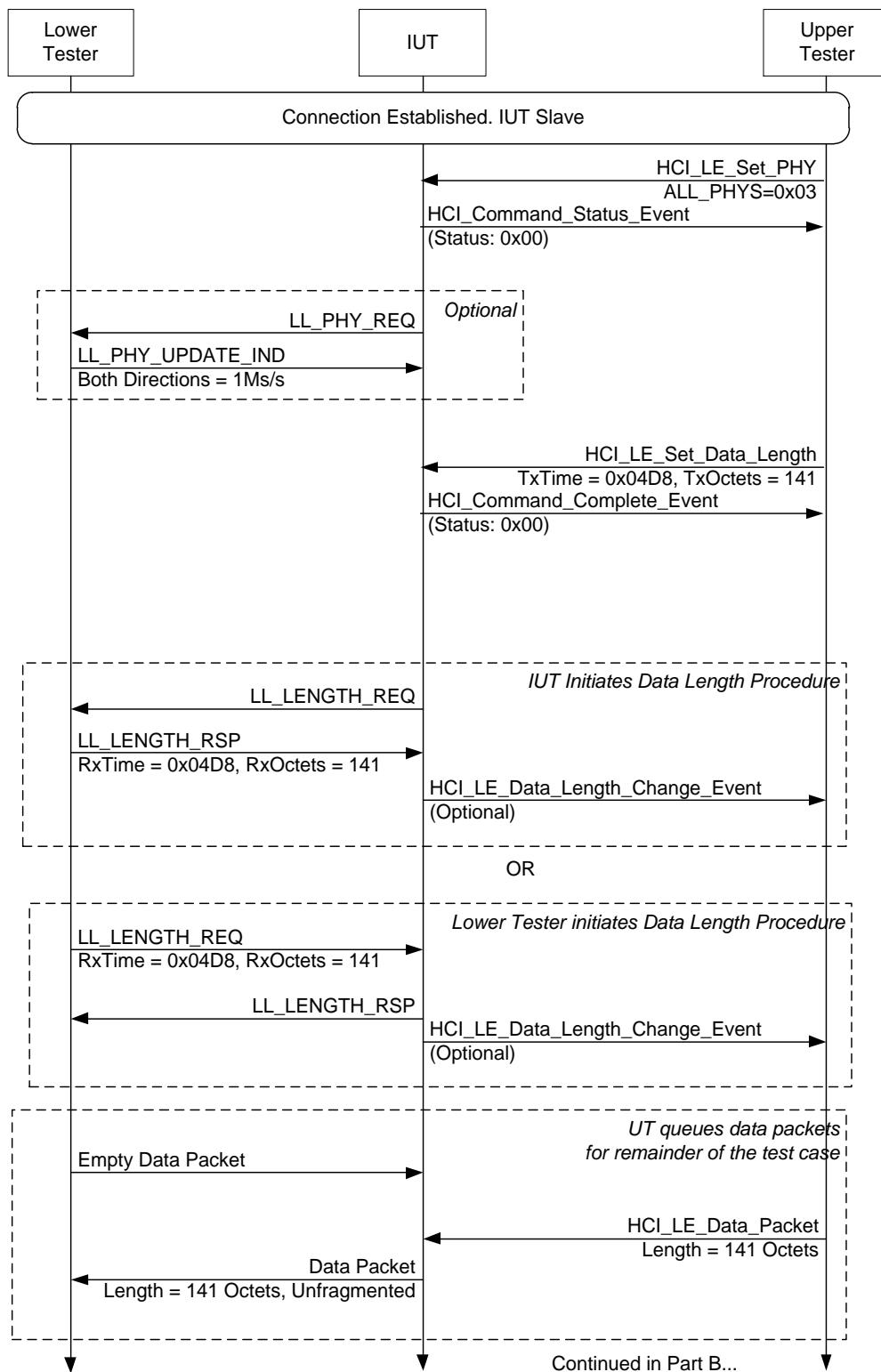


Figure 4.194: LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]



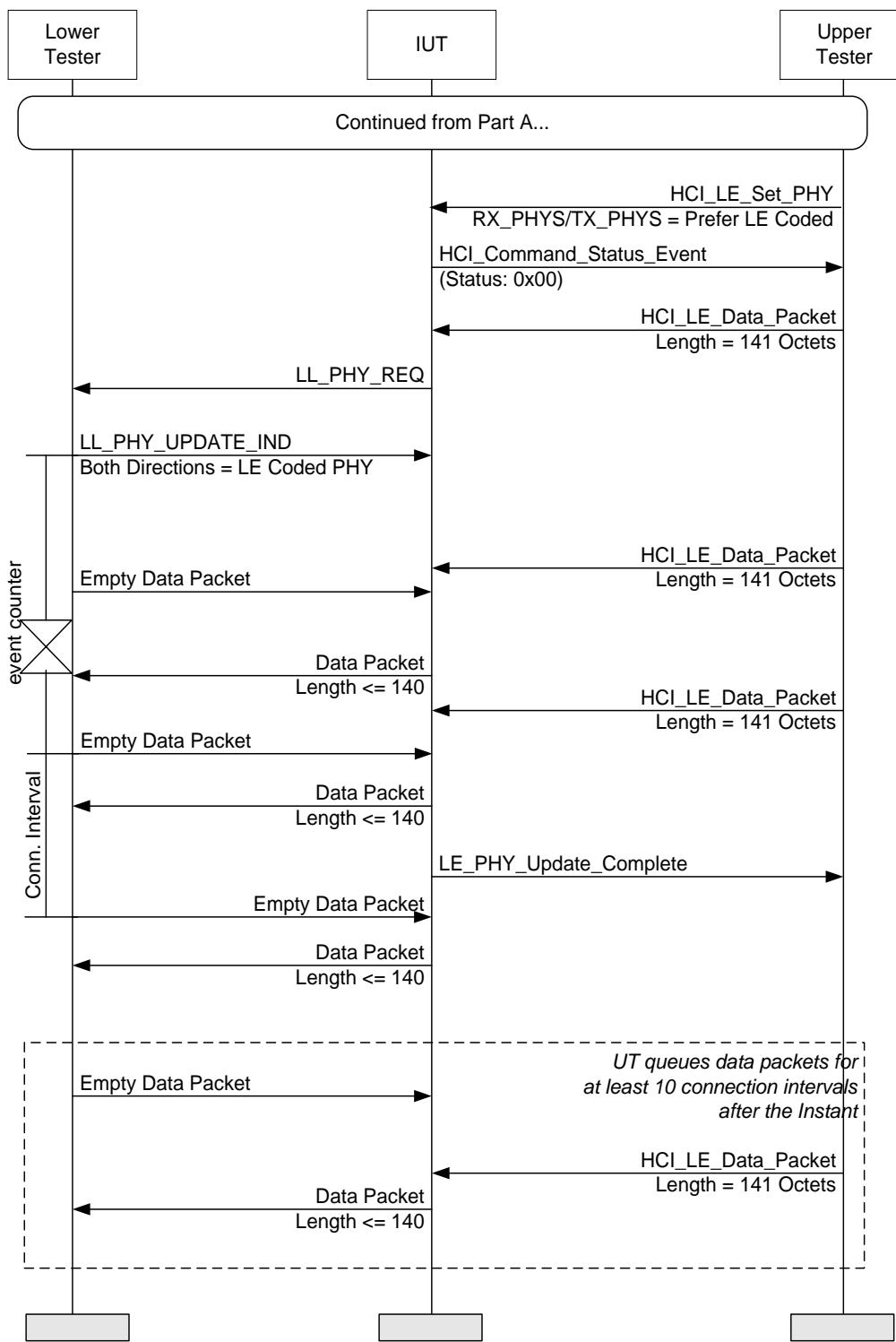


Figure 4.195: LL/CON/SLA/BV-55-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
 2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
 4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
 5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHYS and TX_PHYS both set to prefer LE Coded PHY.
 8. The Lower Tester expects an LL_PHY_REQ from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
 9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 11 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY Update Procedure.
 10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.
 12. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.
 13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in Step 8 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in Step 8 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and



connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One of more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in Step 8.
- The IUT does not allow the Lower Tester to select the LE Coded PHY in Step 8.

4.3.4.45 LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded]

- Test Purpose

Tests that a slave IUT follows packet time restrictions both during and after PHY change when it responds to a PHY Update Procedure from a master. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester, in the master role, maintains a connection.

A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure.

- Reference
[10] 5.1.10.1
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

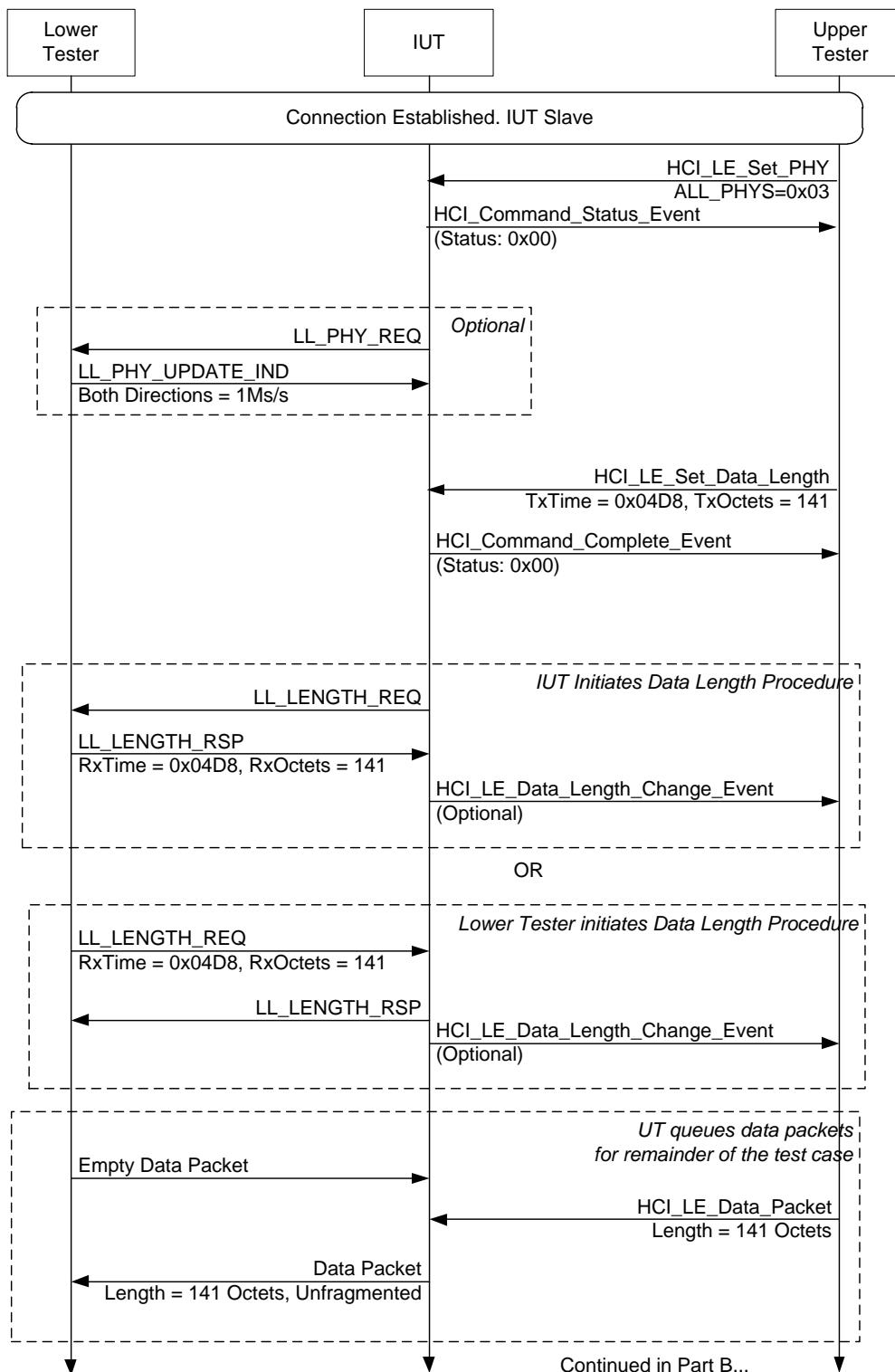


Figure 4.196: LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]



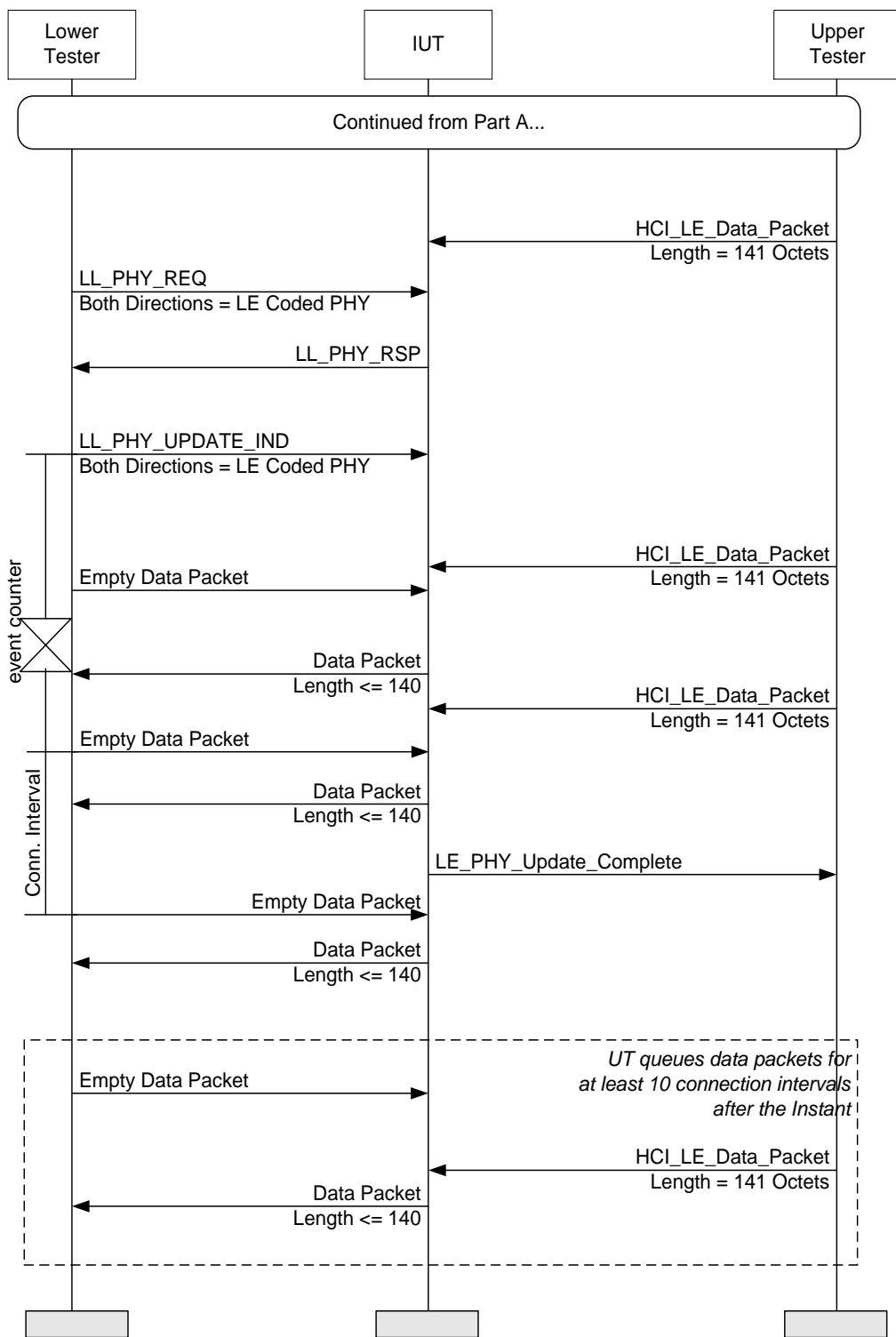


Figure 4.197: LL/CON/SLA/BV-56-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.



3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
 4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
 5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with a fail verdict.
 7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.
 8. The Lower Tester expects an LL_PHY_RSP from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
 9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 11 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY Update Procedure.
 10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.
 12. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.
 13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in Step 8 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in Step 8 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and



connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in Step 8.
- The IUT does not allow the Lower Tester to select the LE Coded PHY in Step 8.

4.3.4.46 LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded]

- Test Purpose

Tests that a slave IUT still transmits data even when the TxTime and/or RxTime values for LE Coded PHY suggest a smaller possible data length than the minimum length data PDU (27 octets).

The Lower Tester, in the master role, maintains a connection.

If a Data Length Update Procedure is performed, minimum settings are used. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure.

- Reference

[10] 5.1.9

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

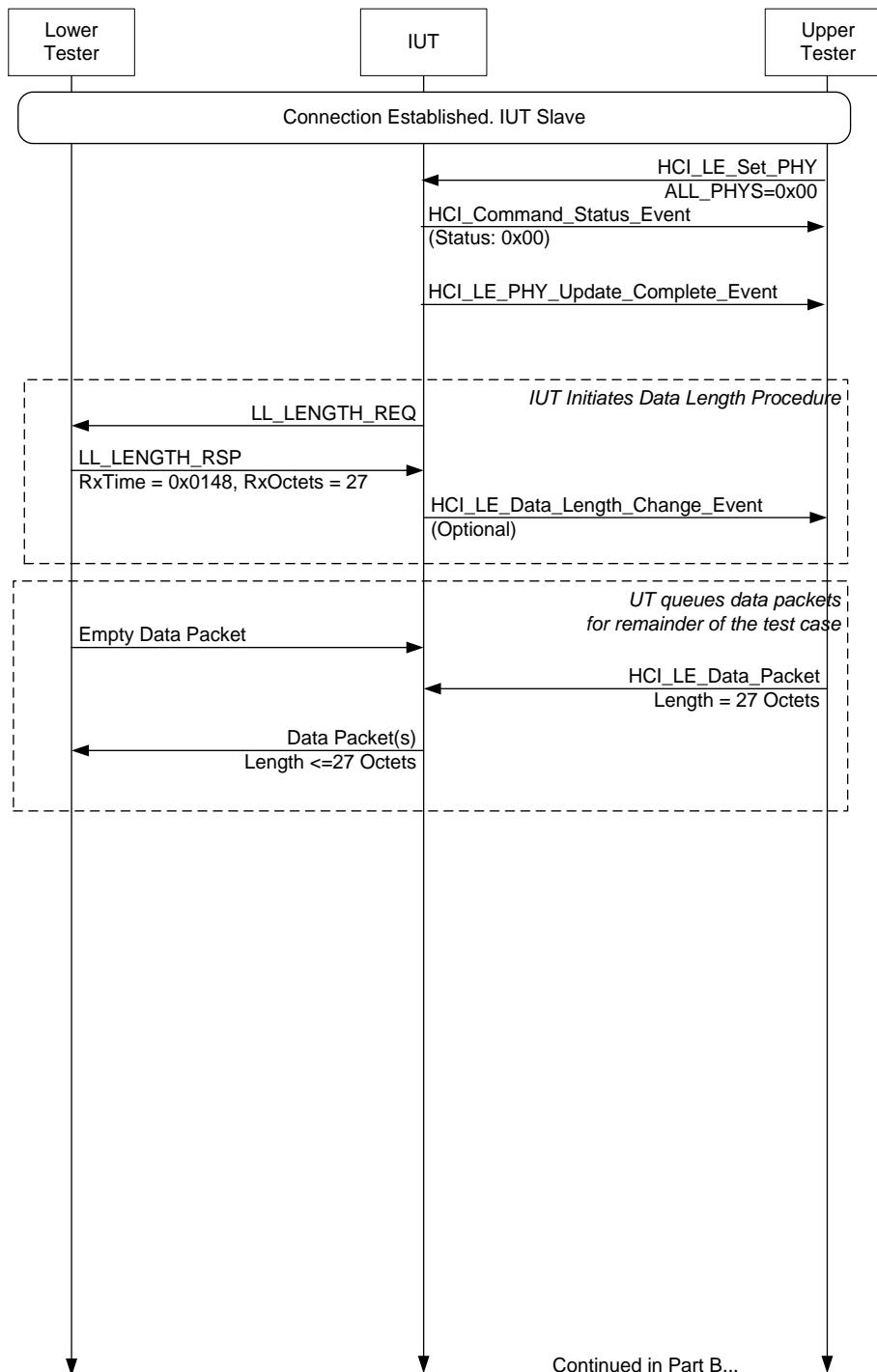


Figure 4.198: LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded – Part A]



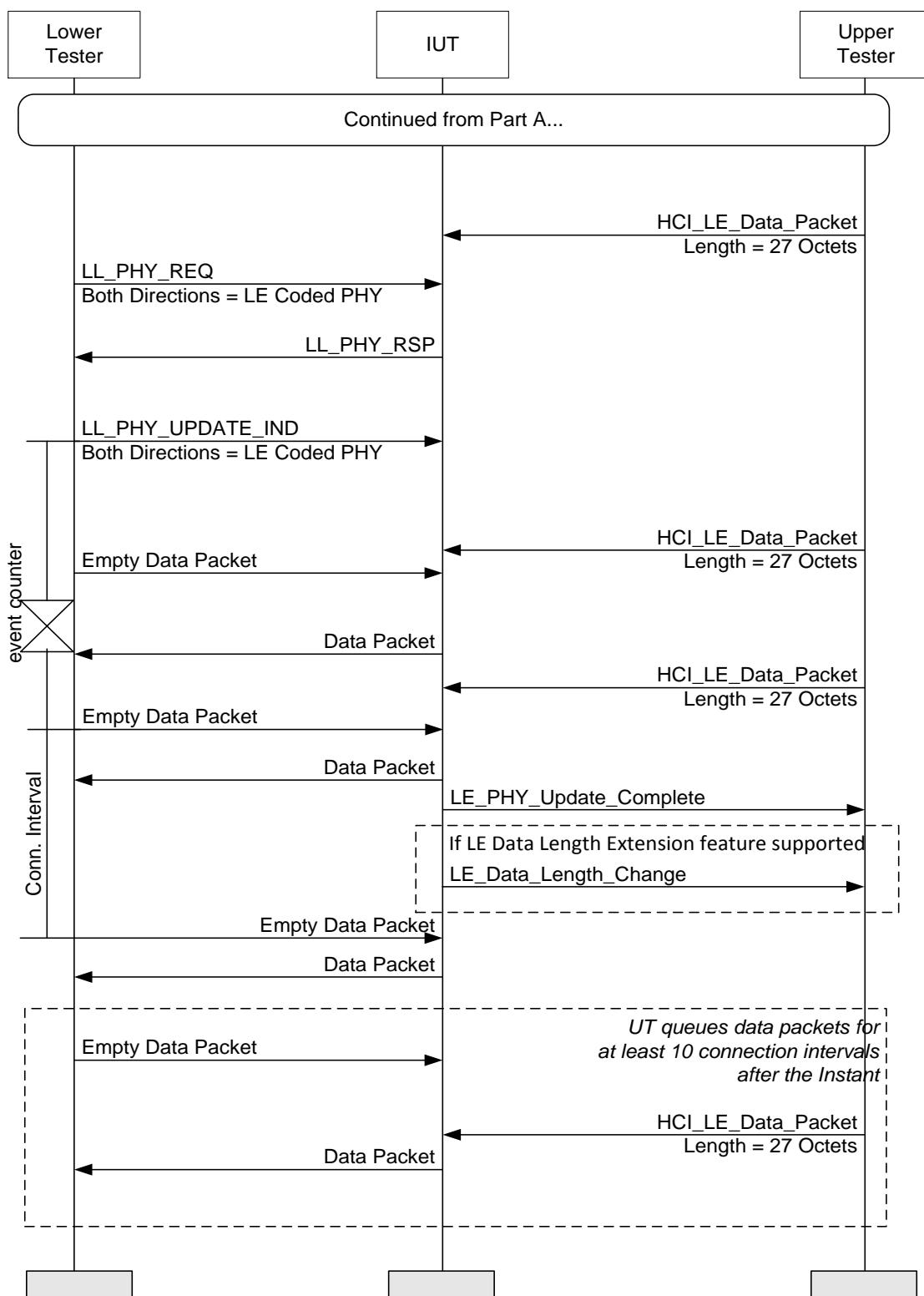


Figure 4.199: LL/CON/SLA/BV-57-C [Mandatory Minimum PDU Length, LE Coded - Part B]

1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with **ALL_PHYS** set to 0x03 and **PHY_options** set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
2. The Upper Tester expects an **HCI_LE_PHY_Update_Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.



3. Optionally, if the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 328 and RxOctets set to 27.
 4. The Upper Tester begins to queue data packets to the IUT with a length of 27 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 5. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.
 6. The Lower Tester expects an LL_PHY_RSP from the IUT which allows for selection of the LE Coded PHY. Lower Tester responds with an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be less than 32767 and greater than 6. If the IUT does not allow selection of the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
 7. The Lower Tester expects that the data packets received are less than or equal to 27 octets in length.
 8. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 27 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 9. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.
 10. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY and, if LE Data Packet Length Extension feature is supported by the IUT, an HCI_LE_Data_Length_Change event with MaxTxOctets and MaxRxOctets set to 27 and MaxTxTime and MaxRxTime set to 2704. These events may be in either order.
 11. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are less than or equal to 27 octets in length. This data exchange continues for at least 10 connection events after the Instant.
 12. Repeat steps 5 to 11, except that in step 5, change 0x04 to 0x01; in steps 6, 9, and 10, change the LE Coded PHY to the LE 1M PHY; and in step 10, change 2704 to 328.
- Expected Outcome

Pass Verdict

Data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP (in Steps 6 and 11) must have a length of 27 octets or less.

The IUT must transition to the LE Coded PHY and LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

If LE Data Packet Length Extension feature is supported by the IUT, the Upper Tester receives HCI_LE_Data_Length_Change events in step 10 with the correct values.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

The IUT does not allow the Lower Tester to select the required PHY in Step 6.



4.3.4.47 LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]

- **Test Purpose**

Tests that a slave IUT follows all packet time restrictions when a PHY Update Procedure is initiated but no PHY change occurs. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant, even if no change occurs.

The Lower Tester acts in the master role maintaining a connection.

A Data Length Update procedure is performed. The Upper Tester begins queuing data to the IUT then issues the HCI command to start the PHY Update Procedure as the Host of the IUT.

- **Reference**

[10] 5.1.10.1

- **Initial Condition**

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

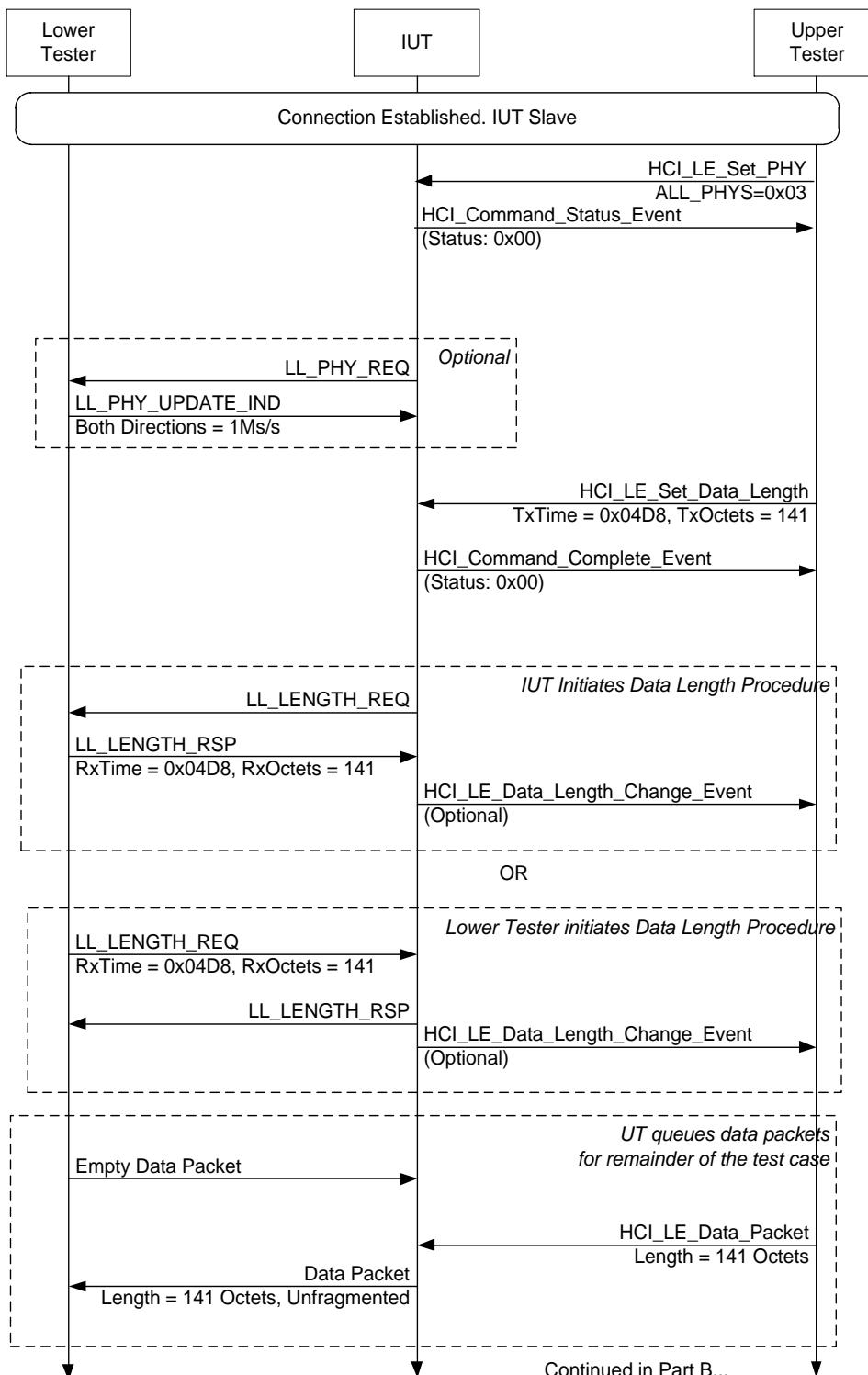


Figure 4.200: LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded – Part A]



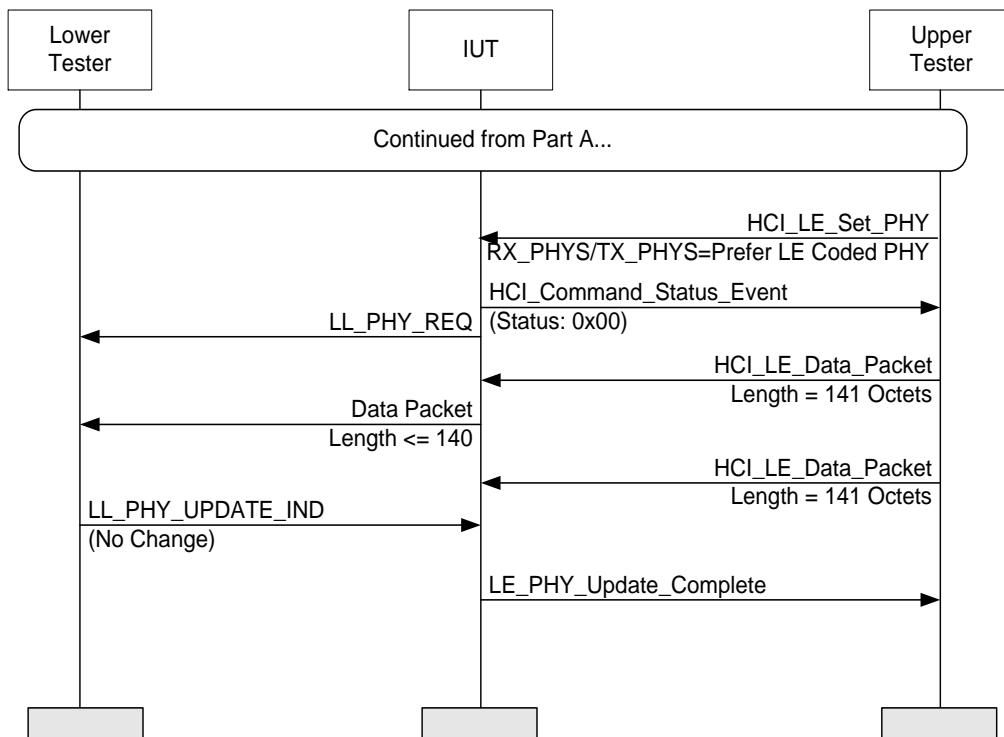


Figure 4.201: LL/CON/SLA/BV-58-C [Initiating PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
4. If the IUT initiates a Data Length Update procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update procedure, the Lower Tester shall initiate the Data Length Update procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHYS and TX_PHYS both set to prefer the LE Coded PHY.
8. The Lower Tester expects an LL_PHY_REQ from the IUT in which the TX_PHYS field has the 0x04 bit set. If the TX_PHYS field does not have the 0x04 bit set, the test ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 423 octets of data are received.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_REQ in step 8 must be 140 octets or less in length. If a larger packet is received during this period, the test case ends with a failed verdict.
 10. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).
 11. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_REQ in Step 8) must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_REQ in Step 8) must have a length of less than or equal to 140 octets, until the Lower Tester sends the LL_PHY_UPDATE_IND in Step 10.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_REQ in Step 8.
- The IUT does not allow the Lower Tester to select the LE Coded PHY in Step 8.

4.3.4.48 LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded]

- Test Purpose

Tests that a slave IUT follows all packet time restrictions both during and after PHY change when it responds to a PHY Update Procedure from a master device but no PHY change occurs. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant, even when no change occurs.

The Lower Tester, in the master role, maintains a connection. A Data Length Update procedure is performed. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure but does not allow it to result in a PHY change.

- Reference

[10] 5.1.10.1



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

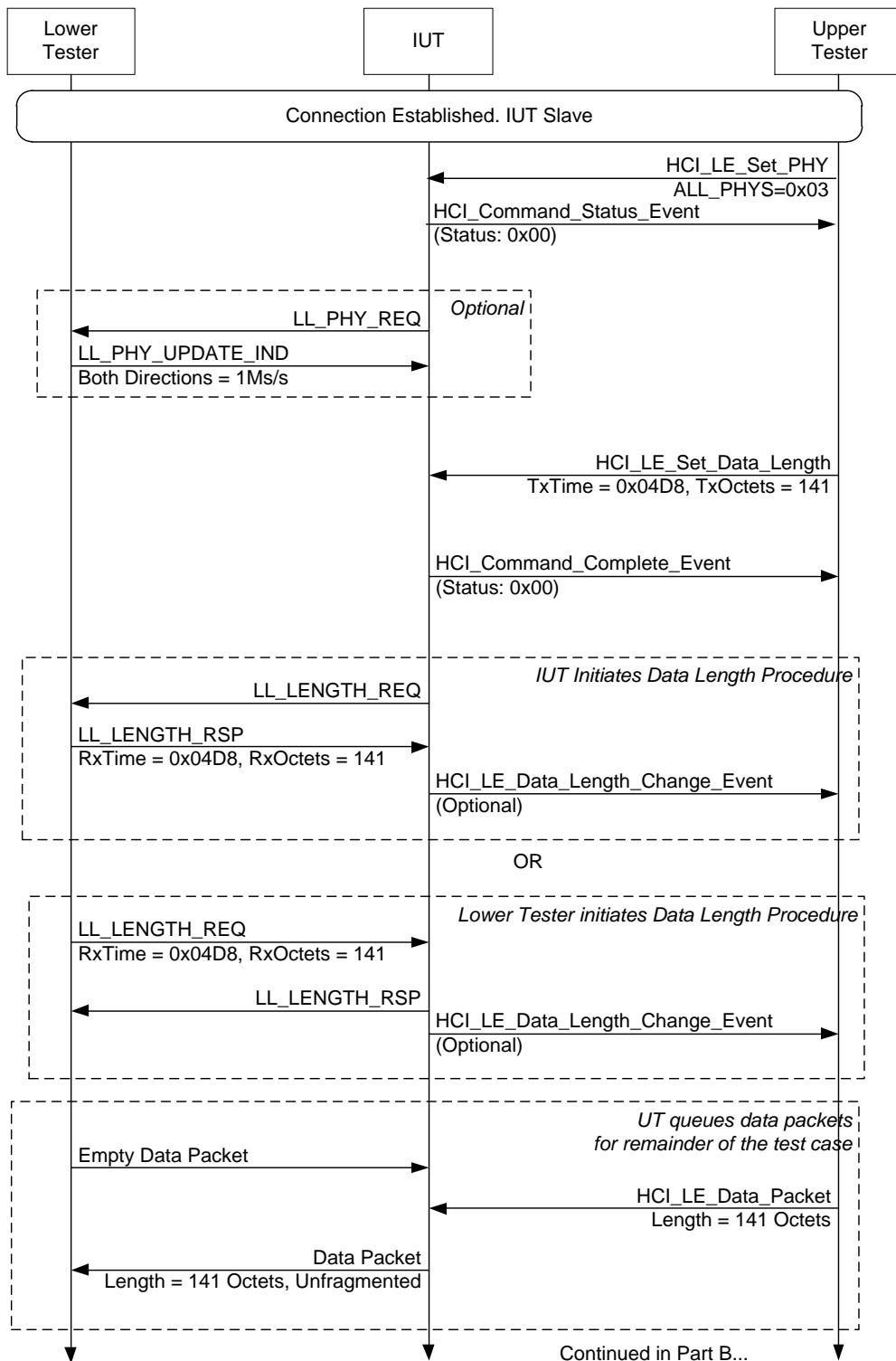


Figure 4.202: LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded – Part A]



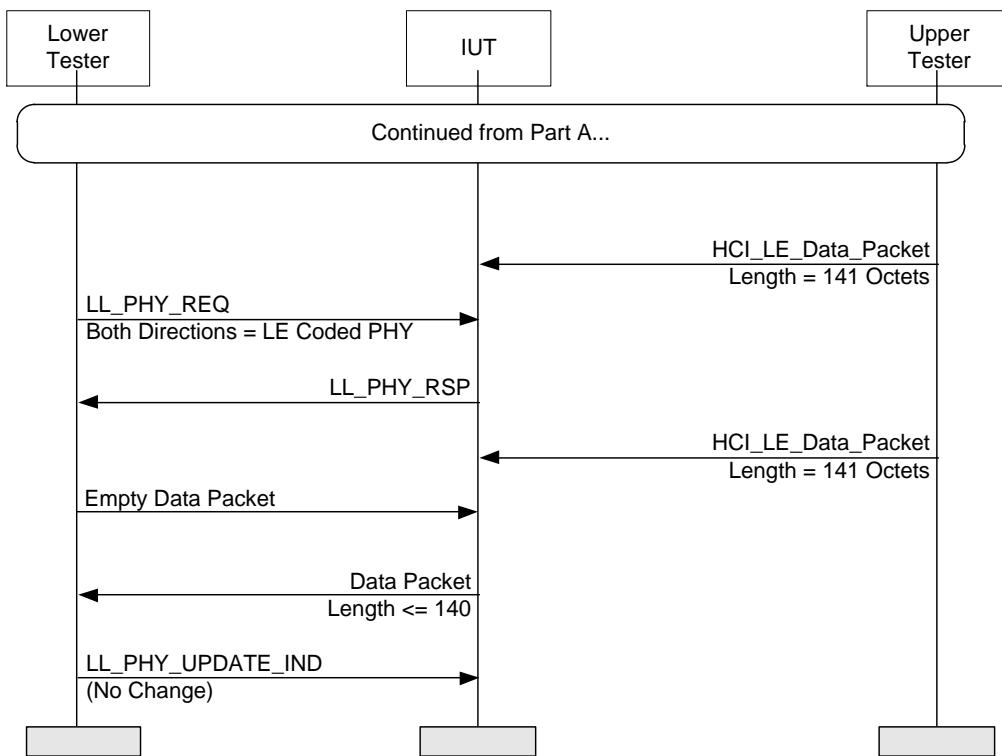


Figure 4.203: LL/CON/SLA/BV-59-C [Responding to PHY Update Procedure – Packet Time Restrictions, No Change, LE Coded - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
4. If the IUT initiates a Data Length Update procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update procedure, the Lower Tester shall initiate the Data Length Update procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x05.
8. The Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). The LE Coded PHY bit must be set in the TX_PHYS field by the IUT or the test case ends with an Inconclusive Verdict. The Lower Tester continues to receive packets from the IUT and delays sending an LL_PHY_UPDATE_IND until at least 423 octets are received.

9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_RSP in step 8 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict.
10. Lower Tester responds with an LL_PHY_UPDATE_IND selecting no change to the PHY in either direction (all fields zero).
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_RSP in Step 8 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_RSP in Step 8 must have a length of less than or equal to 140 octets until the Lower Tester sends the LL_PHY_UPDATE_IND in Step 10.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_RSP in Step 8.
- The IUT does not set the LE Coded PHY bit in the TX_PHYS field in Step 8.

4.3.4.49 LL/CON/SLA/BV-77-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]

This test is identical to LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY] except that the IUT is in the Slave role.

4.3.4.50 LL/CON/SLA/BV-78-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]

This test is identical to LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY] except that the IUT is in the Slave role.

4.3.4.51 LL/CON/SLA/BV-79-C [Slave Data Length Update – Master does not support; LE 1M PHY]

This test is identical to LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY] except that the IUT is in the Slave role.

4.3.4.52 LL/CON/SLA/BV-80-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]

This test is identical to LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY] except that the IUT is in the Slave role.



4.3.4.53 LL/CON/SLA/BV-81-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY]

This test is identical to [LL/CON/MAS/BV-77-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY\]](#) except that the IUT is in the Slave role.

4.3.4.54 LL/CON/SLA/BV-82-C [Slave Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY]

This test is identical to [LL/CON/MAS/BV-78-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY\]](#) except that the IUT is in the Slave role.

4.3.4.55 LL/CON/SLA/BV-83-C [Slave Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY]

This test is identical to [LL/CON/MAS/BV-79-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY\]](#) except that the IUT is in the Slave role.

4.3.4.56 LL/CON/SLA/BV-84-C [Slave Data Length Update – Master does not support; LE Coded PHY]

This test is identical to [LL/CON/MAS/BV-80-C \[Master Data Length Update – Slave does not support; LE Coded PHY\]](#) except that the IUT is in the Slave role.

4.3.4.57 LL/CON/SLA/BV-85-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when a feature exchange has not been performed and the remote device does not support the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and sends an LL_UNKNOWN_RSP PDU in response to the IUT's request.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than

LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

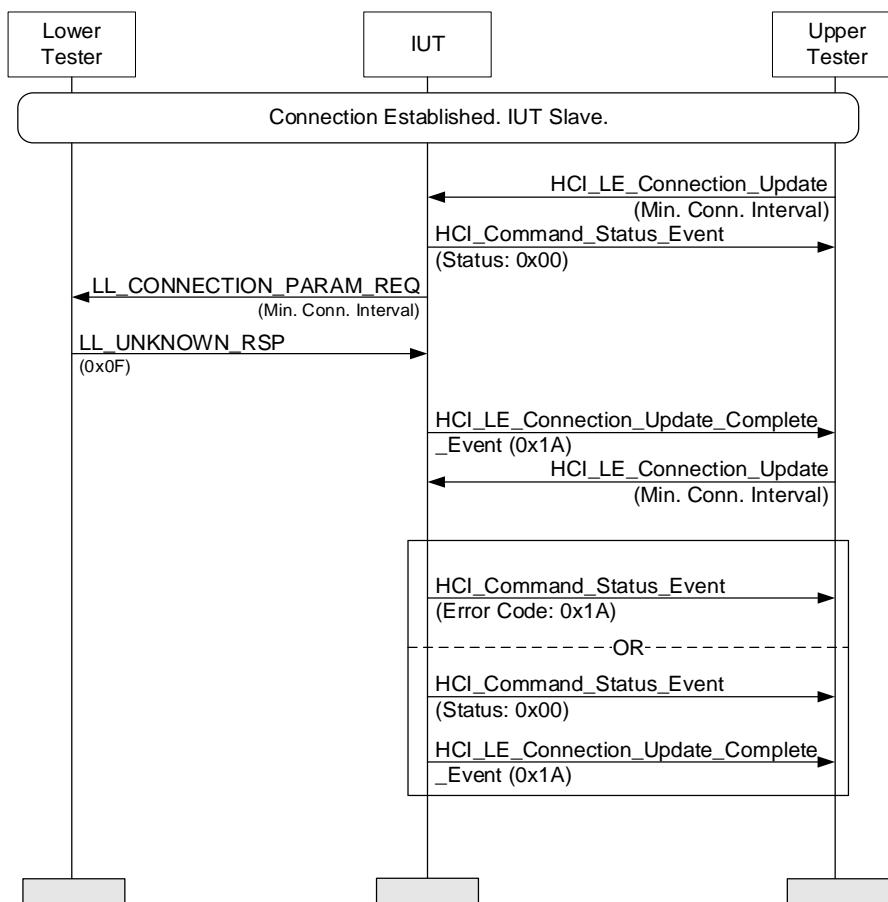


Figure 4.204: LL/CON/SLA/BV-85-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]

1. If the IUT autonomously initiates a feature exchange before step 3, the test ends with an Inconclusive Verdict.
2. Upper Tester sends an HCI_Le_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
3. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester) and the Lower Tester responds with an LL_UNKNOWN_RSP PDU with the Opcode field set to LL_CONNECTION_PARAM_REQ (0x0F).
4. Upper Tester expects an HCI_Le_Connection_Update_Complete event from the IUT containing the error code Unsupported Remote Feature (0x1A).
5. Upper Tester resends the same HCI_Le_Connection_Update command to the IUT and expects the IUT to respond with either:
 - a. an HCI_Command_Status event containing the error code Unsupported Remote Feature (0x1A), or
 - b. an HCI Command Status event indicating success and an HCI_Le_Connection_Update_Complete event containing the error code Unsupported Remote Feature (0x1A).

If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update the connection parameters.

The IUT reports an HCI_LE_Connection_Update_Complete event containing the error code Unsupported Remote Feature (0x1A).

The IUT rejects the second command with the Unsupported Remote Feature (0x1A) error code.

4.3.4.58 LL/CON/SLA/BV-86-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

- Test Purpose

Test that a slave IUT is able to reject the connection parameter request procedure after the feature exchange reveals that the remote device does not support the request.

The Lower Tester acts in the master role maintaining a connection and initiates feature exchange, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
 LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

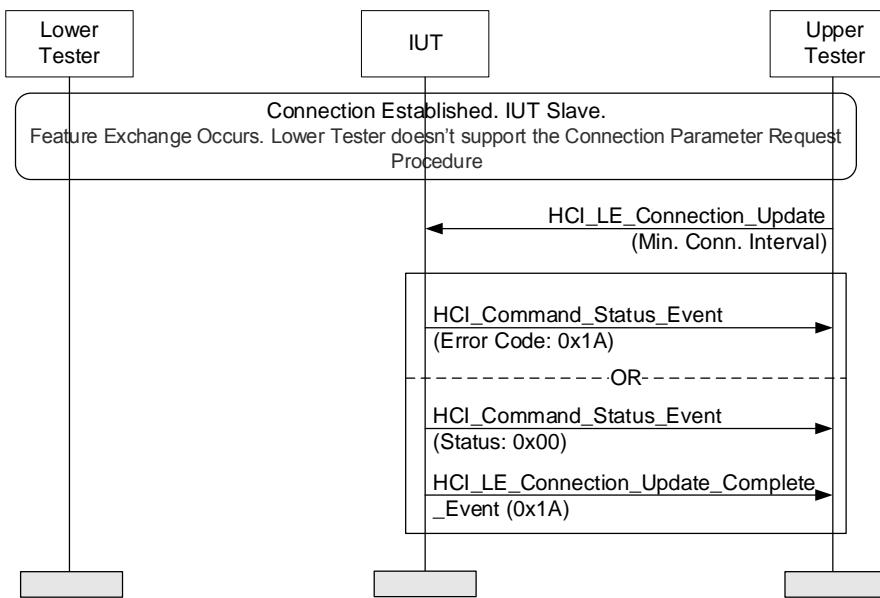


Figure 4.205: LL/CON/SLA/BV-86-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

1. Lower Tester initiates a feature exchange, unless the IUT has already done so. Lower Tester indicates that it does not support the Connection Parameter Request Procedure.
2. Upper Tester sends an HCI_CONNECTION_UPDATE command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects the IUT to respond with either:
 - a. an HCI_COMMAND_STATUS_EVENT containing the error code Unsupported Remote Feature (0x1A), or
 - b. an HCI_COMMAND_STATUS_EVENT indicating success and an HCI_CONNECTION_UPDATE_COMPLETE_EVENT containing the error code Unsupported Remote Feature (0x1A).

If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.

- Expected Outcome

Pass Verdict

The IUT rejects the command with the Unsupported Remote Feature (0x1A) error code.

4.3.4.59 LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]

- Test Purpose

Test that a slave IUT accepts the master transmission at the beginning of an event as the anchor point, irrespective of the checksum result.

The Lower Tester acts in the master role, starting all events after connection setup with packets with invalid checksums and observes the responses and packet contents from the IUT.

- Reference

[3] 4.5.5



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, Lower Tester address, common connection interval, common slave latency, common channel map, any SCA value).

- Test Procedure

Execute the test procedure using a connection timeout value of 3000ms. The test uses the common current and next variables for SN and NESN.

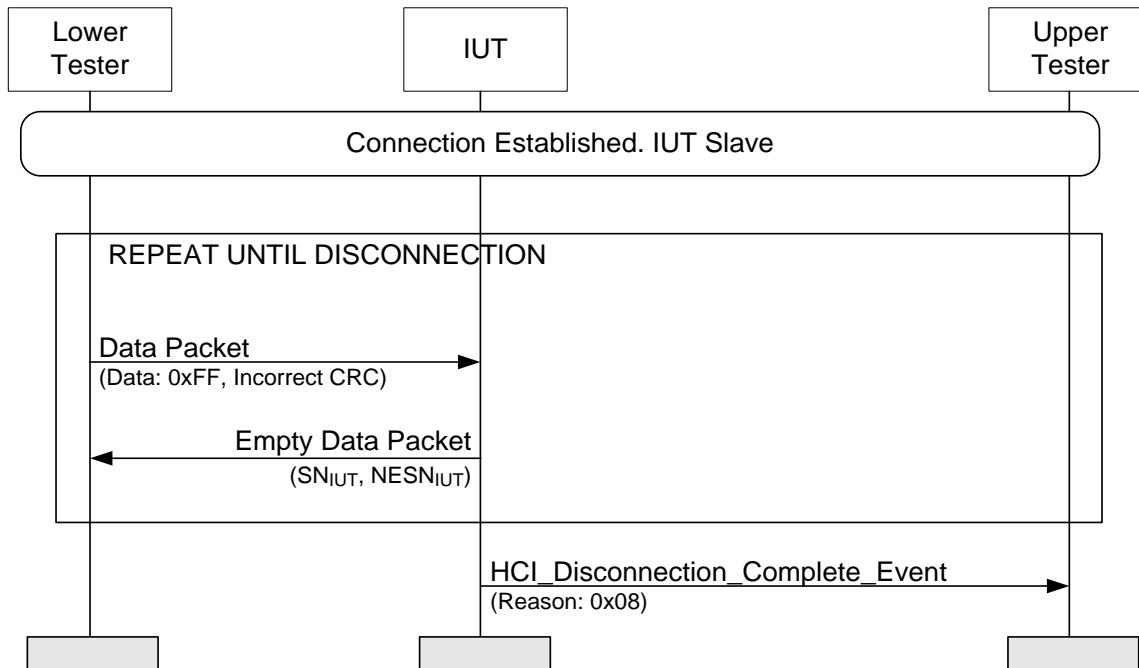


Figure 4.206: LL/CON/SLA/BI-01-C [Invalid CRC Anchor Point]

- Lower Tester sends a DATA packet with an invalid checksum once a connection interval to the IUT, using the data channel selection parameters.
- Lower Tester expects a DATA packet after TIFS in events where the IUT is required to listen. Allow mismatches for packets not received correctly.
- Repeat steps 1-2 for a period longer than the connection supervision timeout value:
- Upper Tester expects an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating connection timeout (0x08).

- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.

- Expected Outcome

Pass Verdict

The test procedure executes with the parameters selected,

The IUT responds in at least 65 of the connection events measured (300).



- Notes

The error rate for the preamble and access address for the BER is the same as for test [LL/CON/SLA/BV-10-C \[Accepting Parameter Update\]](#), but the response rate required is divided by slave latency plus 1.

4.3.4.60 LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]

- Test Purpose

Test the correct behavior of a slave IUT when TERMINATE_IND packets are not acknowledged.

The Lower Tester acts in the master role, expecting TERMINATE_IND packets from the IUT and not acknowledging them.

- Reference

[\[3\] 5.1.6](#)

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common connection parameters.

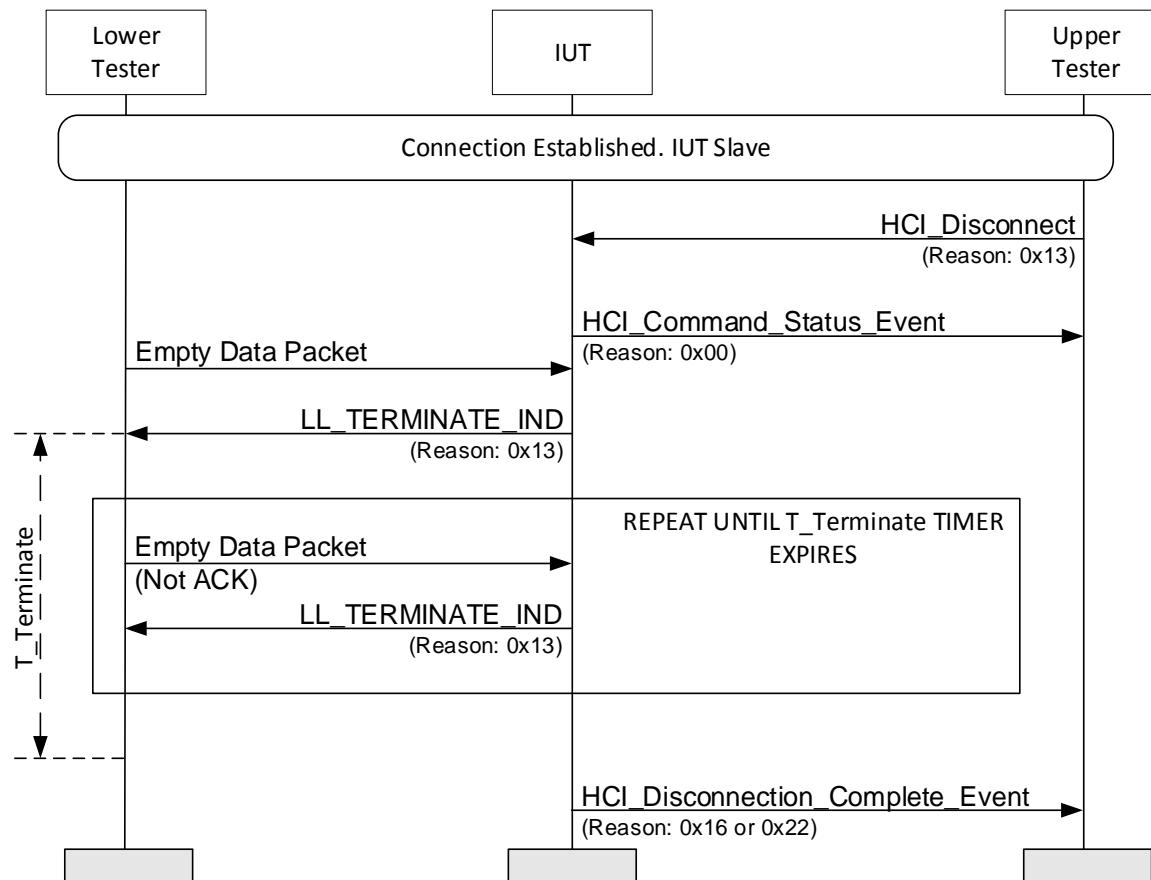


Figure 4.207: LL/CON/SLA/BI-02-C [Slave T_Terminate Timer]

1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status in response.
 2. Lower Tester expects the IUT to respond to a master transmission with a TERMINATE_IND packet. Do not acknowledge the termination or any following packet but continue master transmissions.
 3. Lower Tester sends an empty DATA packet to the IUT not acknowledging any responses. Repeat until T_Terminate timer expires.
 4. Upper Tester expects an HCI_Disconnection_Complete event including status of 0x00 (success), and a reason code of 0x16 ("Connection Terminated by Local Host") or 0x22 ("LL Response Timeout") from the IUT indicating loss of the link.
- Expected Outcome

Pass Verdict

The IUT reports the connection termination with an HCI event.

The IUT keeps sending TERMINATE_IND packets until T_Terminate timer expires.

4.3.4.61 LL/CON/SLA/BI-04-C [Rejecting Connection Change]

- Test Purpose

Test that a slave IUT either terminates the connection or successfully maintains the connection upon receiving a control packet indicating a past event for the change.

The Lower Tester acts in the master role in the connection, transmitting control packets with invalid event counter values and observes the IUT notifying that the connection has been terminated.

- Reference

[3] 5.1.1, 5.1.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value).

IUT is Slave and Lower Tester is Master.

- Test Procedure

Execute the test procedure using the common connection parameters.



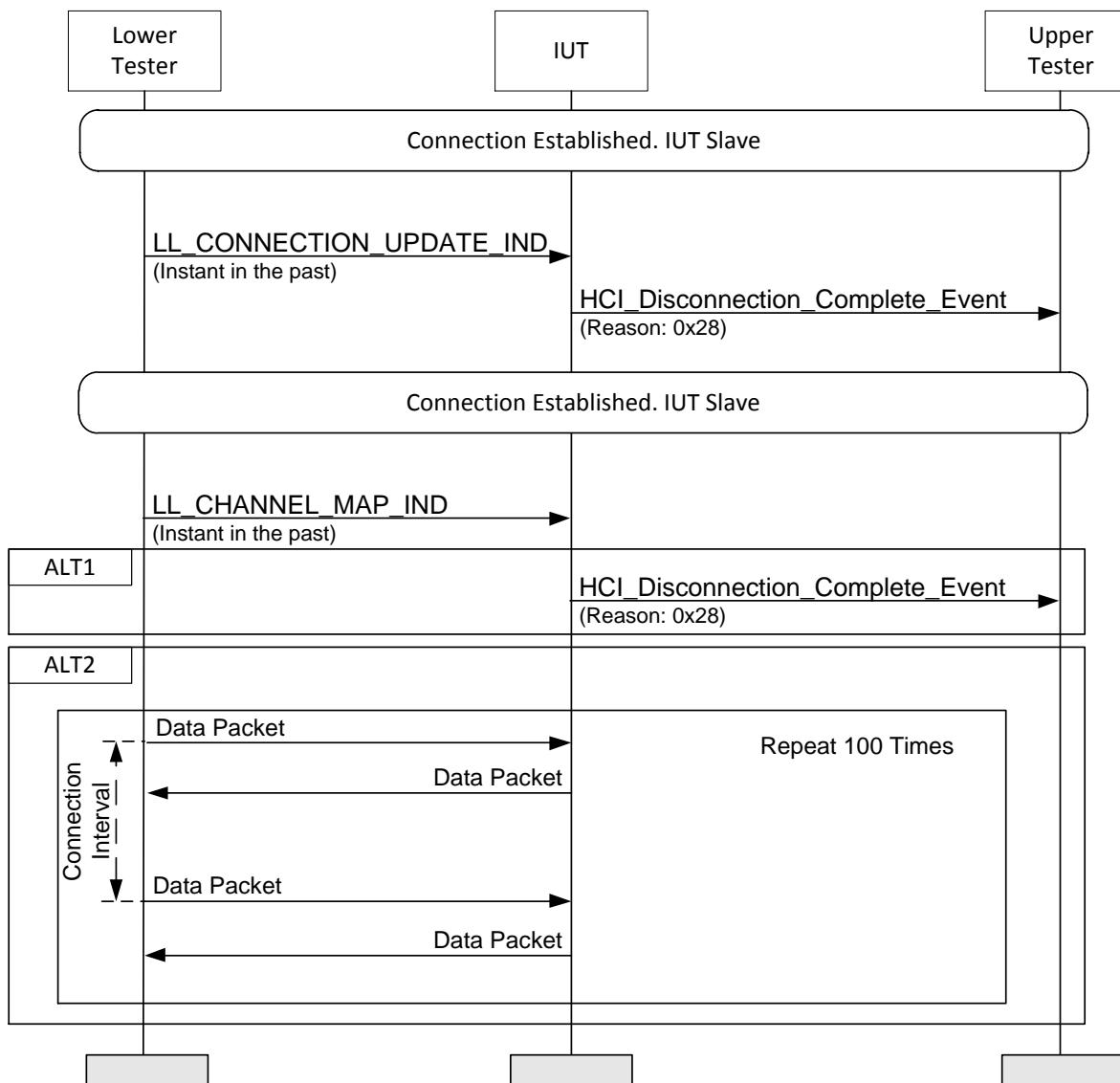


Figure 4.208: LL/CON/SLA/BI-04-C [Rejecting Connection Change]

1. Lower Tester sends an LL CONNECTION_UPDATE_IND packet setting the connection parameters to the intermediate values, but the event counter equal to or less than the present event. The IUT may send a packet acknowledging the connection update request.
2. Upper Tester expects an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating instant passed (0x28).
3. A connection is established
4. Lower Tester sends an LL CHANNEL_MAP_IND packet with the event counter equal to or less than the present event to the IUT using the first selected channel map.

Alternative 1:

- The IUT may send an acknowledgement.
- The Upper Tester expects an HCI_Disconnection_Complete event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating instant passed (0x28).
- Alternative 2:
- The IUT sends an acknowledgement.



- The IUT does not send an HCI_Disconnection_Complete event to the Upper Tester and maintains the connection with the Lower Tester.
- Expected Outcome

Pass Verdict

If alternative 1 is followed in step 4:

- The IUT stops maintaining the connection, at any time before a period equal to the connection supervision timeout value has passed from the invalid request.
- The IUT reports the connection failure with an HCI event.

If alternative 2 is followed in step 4, the IUT maintains the connection with the Lower Tester and does not report a connection failure.

4.3.4.62 LL/CON/SLA/BI-05-C [Slave Connection Control Timer]

- Test Purpose

Test that a slave device is able to recover from a control procedure failure.

The Lower Tester acts in the master role, starting and interrupting a control procedure in order for the slave connection control timer to expire.

The host shall be informed about the failure of a control procedure if the link is not disconnected before the completion of the control procedure.

The host should not be informed about the failure of a control procedure if the link is disconnected before the completion of the control procedure.

- Reference

[3] 5.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure

Execute the test procedure using the common connection parameters.



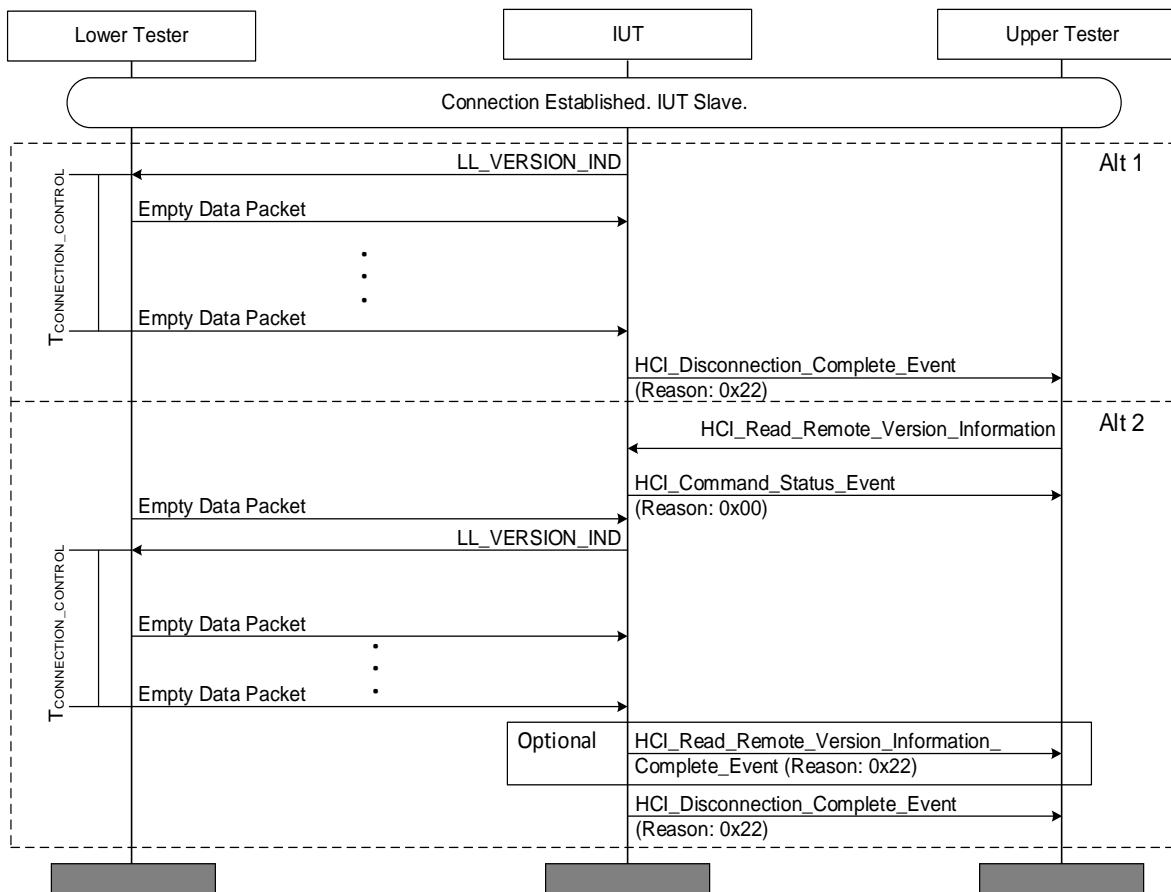


Figure 4.209: LL/CON/SLA/BI-05-C [Slave Connection Control Timer]

- Once connection is established, if the IUT has sent LL_VERSION_IND packet by itself, the Lower Tester acknowledges the version packet. The Lower Tester continues the master transmissions but it never sends the LL_VERSION_IND packet. Then, continue on step 6.
 - If the IUT does not send LL_VERSION_IND packet by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status_Event in response.
 - The Lower Tester expects the IUT to respond to a master transmission with a VERSION_IND packet. The Lower Tester acknowledges the version packet.
 - The Lower Tester sends empty DATA packets to the IUT until the connection control timeout value, or until Step 5 executes.
 - The IUT sends to the Upper Tester the event HCI_Read_Remote_Version_Information_Complete_Event indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from LL response timeout.
 - The Upper Tester expects an HCI_Disconnection_Complete_Event from the IUT indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from connection control transaction timeout.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping to respond to the master transmissions,



The IUT reports the connection termination with an HCI event.

4.3.4.63 LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]

- Test Purpose

Test that a slave IUT is able to perform the connection parameter request procedure when the remote device does not respond to the request.

The Lower Tester acts in the master role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and does not respond to the IUT's request.

- Reference

[3] 5.1.7

- Initial Conditions

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

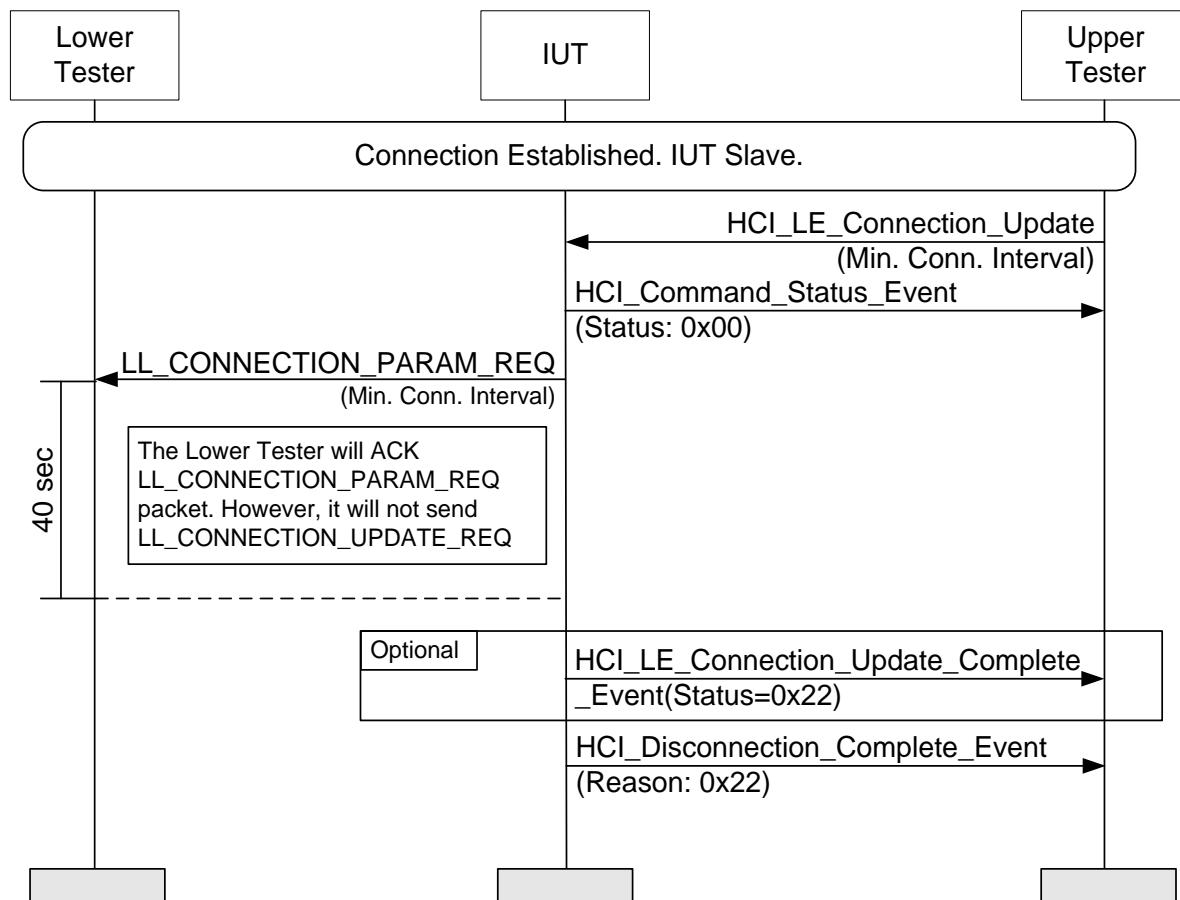


Figure 4.210: LL/CON/SLA/BI-07-C [Initiating Connection Parameter Request – Timeout]



1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
 2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). The Lower Tester acknowledges the LL_CONNECTION_PARAM_REQ PDU but does not respond to the LL_CONNECTION_PARAM_REQ.
 3. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.
- Expected Outcome

Pass Verdict

The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

4.3.4.64 LL/CON/SLA/BI-08-C [Accepting Connection Parameter Request – Illegal Parameters]

- Test Purpose

Test that a slave IUT is able to respond to a connection parameter request procedure from a master device when the connection parameter request from the master contains illegal parameters.

- Reference

[3] 5.1.7

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX,
 LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).



- Test Procedure

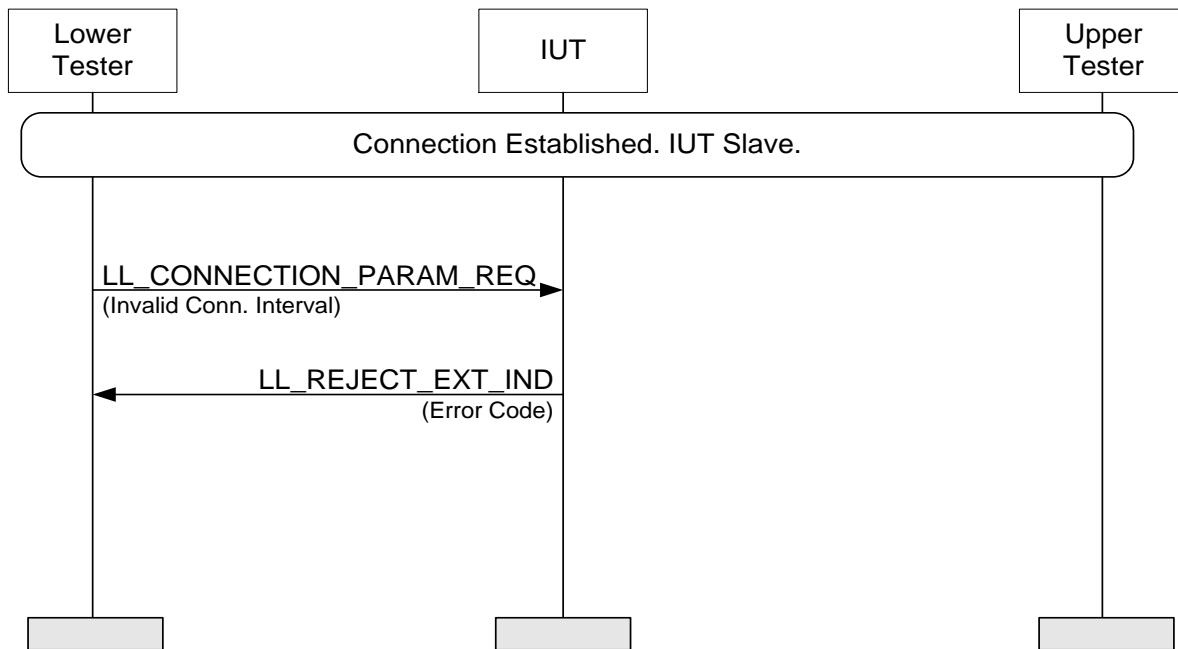


Figure 4.211: LL/CON/SLA/BI-08-C [Accepting Connection Parameter Request – illegal parameters]

1. Lower Tester sends an **LL_CONNECTION_PARAM_REQ** PDU to the IUT setting the connection interval min and max to outside the valid range i.e. 4 (6ms).
 2. Lower Tester expects an **LL_REJECT_EXT_IND** control PDU from the IUT with ErrorCode 0x1E.
- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an **LL_REJECT_EXT_IND** using the correct ErrorCode.

4.3.4.65 LL/CON/SLA/BI-09-C [Responding to PHY Update Procedure – Instant In Past]

- Test Purpose

Tests a slave IUT's ability to cope when a master-initiated PHY update procedure specifies an instant that is in the past.

- Reference

[10] 5.1.10

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to **LL_slave_connSlaveLatency_MAX**, selected timeout, any SCA value).



- Test Procedure

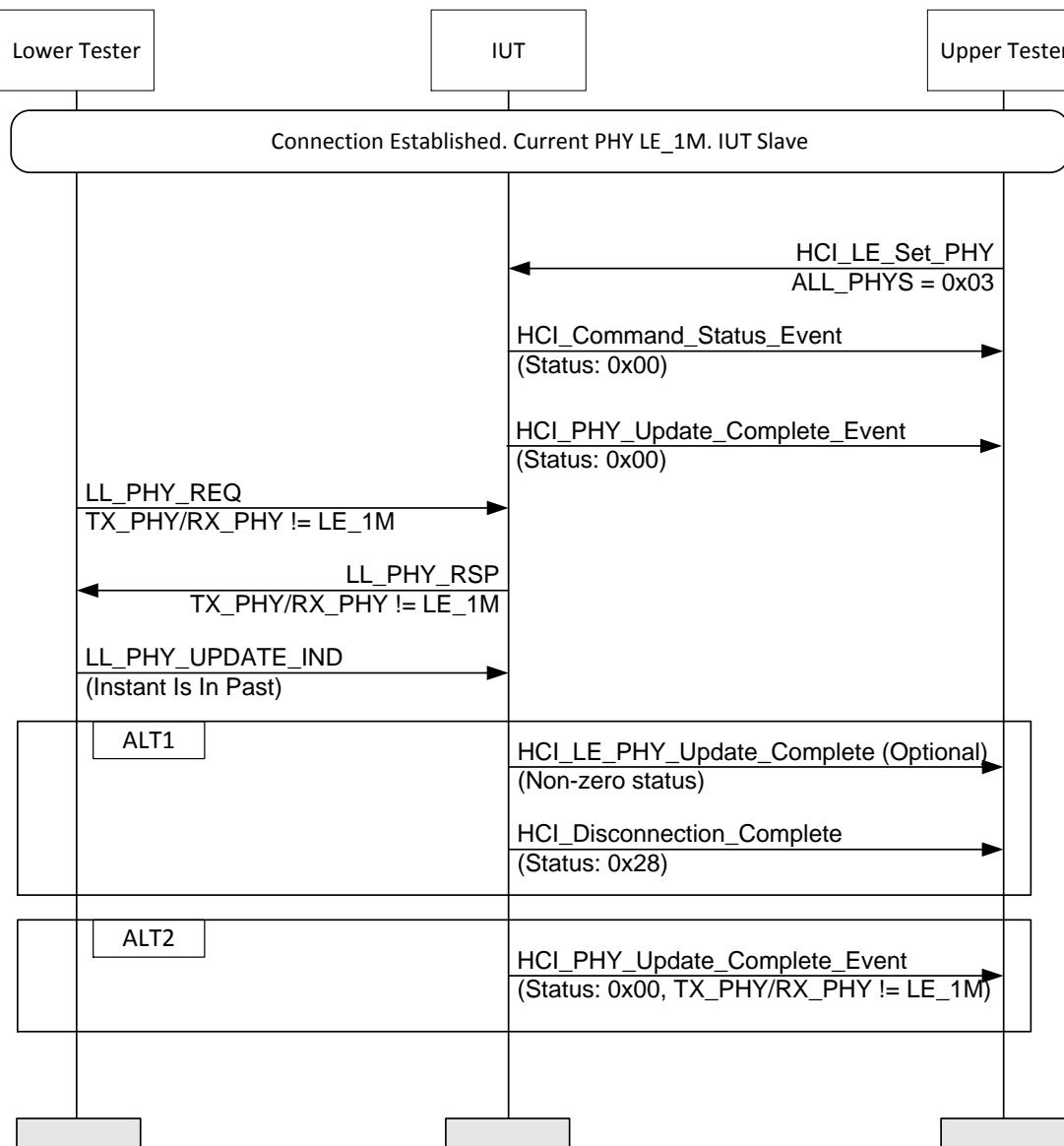


Figure 4.212: LL/CON/SLA/BI-09-C [Responding to PHY Update Procedure – Instant In the Past]

- Upper Tester sends an HCI_LE_Set_PHY command to the IUT with the ALL_PHYS fields set to a value of 0x03. Upper Tester expects an HCI_Command_Status event AND AN HCI_PHY_UPDATE_COMPLETE EVENT indicating success in response.
- The Lower Tester sends an LL_PHY_REQ PDU with TX_PHYS and RX_PHYS both set to prefer a PHY other than LE 1M.
- Lower Tester expects an LL_PHY_RSP control PDU from the IUT with at least one bit set in each field (TX_PHYS, RX_PHYS). A PHY other than LE 1M must be set in both fields by the IUT or the test case ends with an Inconclusive Verdict.
- Lower Tester responds with an LL_PHY_UPDATE_IND PDU with the both the M_TO_S_PHY and S_TO_M_PHY fields set to select a PHY other than LE 1M and with an Instant field set such that (Instant – connEventCount) modulo 65536 shall be greater than or equal to 32767 and completes the procedure.

Alternative 1:

- The Upper Tester optionally expects the IUT to send an HCI_LE_PHY_Update_Complete event with a non-zero status.
- The IUT sends an HCI_Disconnection_Complete_Event (Reason: 0x28) to the Upper Tester and the IUT stops maintaining the connection.

Alternative 2:

- The IUT sends an HCI_LE_PHY_Update_Complete with TX_PHYS and RX_PHYS both set to the PHYs specified in the LL_PHY_UPDATE_IND PDU.
- The IUT maintains the connection with the Lower Tester on those PHYs.

- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's LL_PHY_REQ with an LL_PHY_RSP PDU with at least one bit set for each field (TX_PHYS, RX_PHYS).

If alternative 1 is followed in step 4:

- If the LL_PHY_UPDATE_IND M_TO_S_PHY or S_TO_M_PHY field is non-zero, the IUT sends an HCI_Disconnection_Complete_Event (Reason 0x28) and the IUT stops maintaining the connection.
- If the IUT sends an HCI_LE_PHY_Update_Complete event to the Upper Tester in Step 5 the status shall be non-zero.

If alternative 2 is followed in step 4, the IUT sends an HCI_LE_PHY_Update_Complete with PHYS other than LE 1M, maintains the connection with the Lower Tester and does not report a connection failure.

Inconclusive Verdict

The IUT does not select a PHY other than LE 1M in both fields in step 3).

4.3.5 MAS

Tests that the IUT behaves according to the connection procedures in the master role.

4.3.5.1 LL/CON/MAS/BV-02-C [Events with Slave Latency]

- Test Purpose

Test that a master IUT is able to maintain a connection when the slave using the slave latency mechanism.

The Lower Tester acts as a slave and uses latencies up to the maximum supported by the IUT.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_master_connSlaveLatency_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, up to LL_master_connSlaveLatency_MAX, timeout, common channel map, not encrypted)



- Test Procedure

Execute the test procedure using a connection interval of 32 ms, a latency of 5 events and a connection timeout parameter of 32 s.

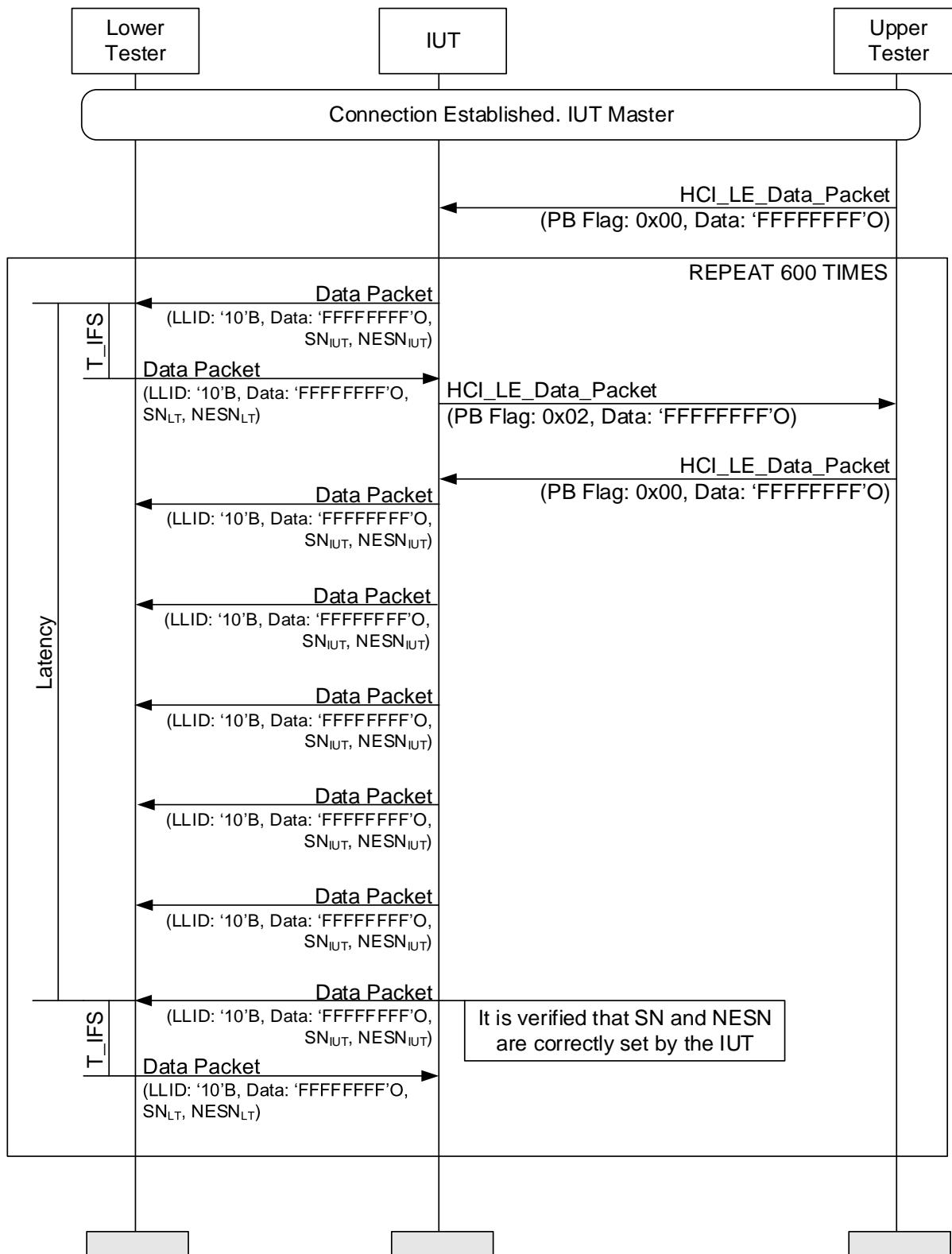


Figure 4.213: LL/CON/MAS/BV-02-C [Events with Slave Latency]

1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0xFFFFFFFF, for a data total length of 4, with Packet_Boundary_Flag flag set.
 2. Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.
 3. Lower Tester sends a DATA packet with in events required by the slave latency parameter only, TIFS after the packet from the IUT. Use only the initial values for SN and NESN.
 4. Upper Tester expects an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 3 and with the Packet_Boundary_Flag flag set
 5. Expect the IUT to continue to repeat the acknowledgement or to indicate that no response was received, with the SN matching the current NESN and the NESN matching the current SN until the Lower Tester sends a DATA packet after slave latency parameter. In case of flow control allow the IUT to updated SN without updating NESN in sent packet.
 6. Repeat steps 1-5 600 times.
 7. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The IUT maintains the connection with slave latency in use,

The IUT observes the acknowledgement scheme taking into account possible flow control in the procedure steps,

The IUT clock drift between the events where the slave is active is acceptable.

- Notes

The connection parameters are selected such that the probability that a connection will be terminated by the connection supervision timer expiring when the slave is using the maximum latency is close to zero. Only the shorter connection interval is tested with slave latency below the maximum possible. When the connection supervision timer is restricted only to $(1 + \text{connSlaveLatency}) * \text{connInterval}$ as in the connection setup parameter constraints, there is still about 0.5 % probability that a connection terminates, which is expected to occur in a geometric distribution about once every 170 tries which is not acceptable for a repeatable test. The probabilities are calculated for empty data packets.

4.3.5.2 LL/CON/MAS/BV-03-C [Master Sending Data]

- Test Purpose

Test that a master IUT is able to send data to a slave device.

The Lower Tester acts in the slave role in the connection, submits data for the master to transmit and observes the data in the packets from the master.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.

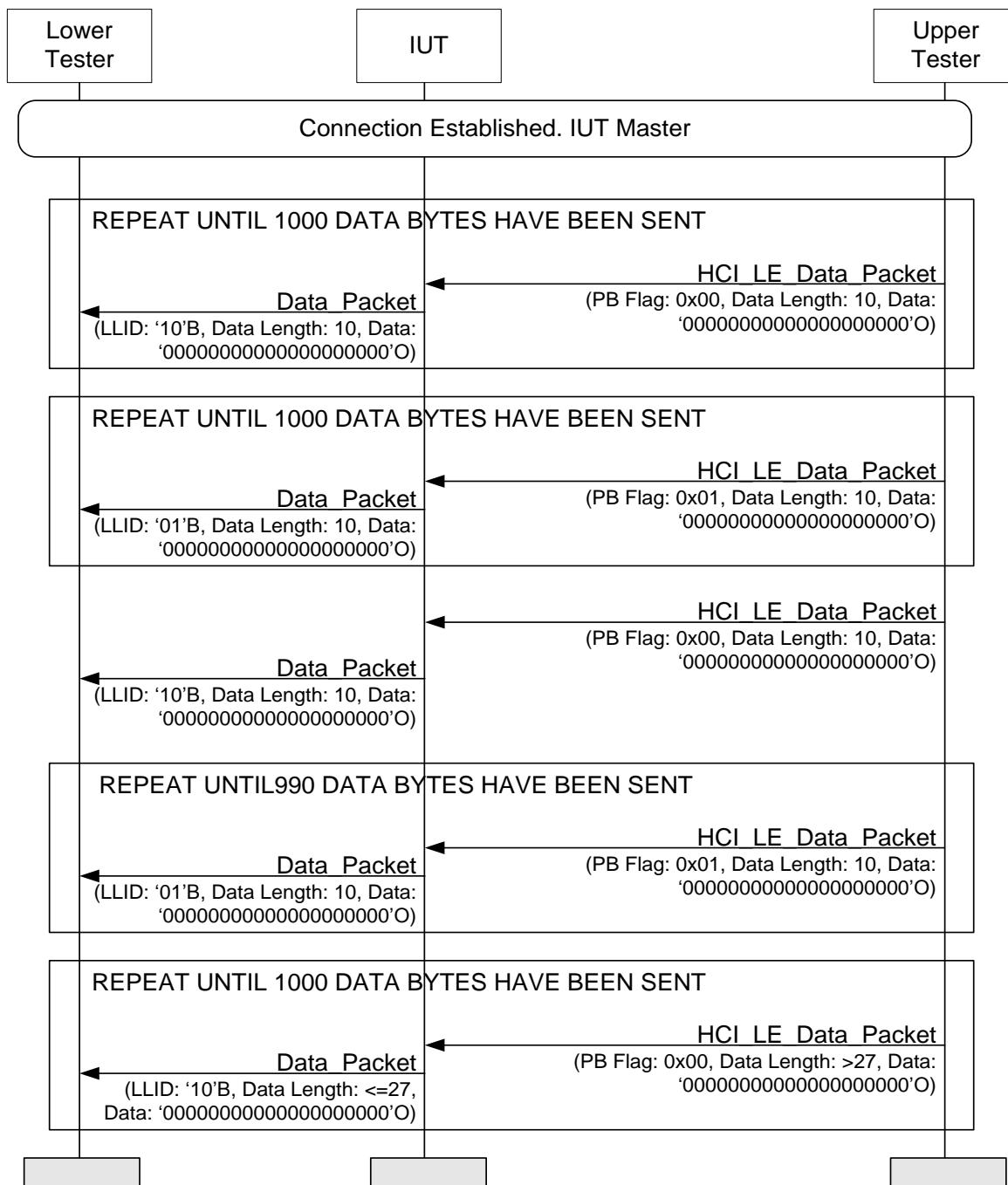


Figure 4.214: LL/CON/MAS/BV-03-C [Master Sending Data]

- Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
- Upper Tester expects the appropriate HCI_Number_Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.



3. Lower Tester expects a DATA packet from the IUT, with the LLID field set to 0x02 and containing a data element submitted in step 1. Lower Tester sends an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel.
4. Repeat steps 2 - 3 until all elements submitted in step 1 have been received.
5. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x01 and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) is successfully submitted.
6. Upper Tester expects the appropriate HCI_Number.Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.
7. Lower Tester expects a DATA packet from the IUT, with the LLID field set to 0x01 and containing a data element submitted in step 5. Send an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel.
8. Repeat steps 6 - 7 until all elements submitted in step 5 have been received.
9. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, with Packet_Boundary_Flag flag equal to 0x00 in the first packet and equal to 0x01 in the following until the selected number of octets (1000) are successfully submitted.
10. Upper Tester expects the appropriate HCI_Number.Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.
11. Lower Tester expects a DATA packet from the IUT, containing a data element submitted in step 9: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following.
12. Lower Tester responds with a DATA packet using the acknowledgement scheme to the IUT on the same data channel.
13. Repeat steps 10-12 until all data sent in step 9 have been reported.

Optional, for devices with a data_packet_length (as defined in [Figure 4.4: Buffer Size Read Preamble Steps](#)) over 27 octets.

14. Optional. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag equal to 0x00 and data elements with the value 0x00, using a data total length from 28 to the data_packet_length (as defined in [Figure 4.4: Buffer Size Read Preamble Steps](#)), until the selected number of octets (1000) are successfully submitted. It expects the appropriate HCI_Number.Of_Completed_Packets events from the IUT using the connection handle, indicating a number of packets completed.
15. Optional. Lower Tester expects a DATA packet from the IUT, containing a data element submitted in step 10: with the LLID field set to 0x00. Send an empty data packet once a connection interval to the IUT using the acknowledgement scheme and the data channel selection parameters.
16. Optional. Repeat step 15 until all elements submitted in step 14 have been received.
17. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble steps).

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT sending all of the data and maintaining the correct sequence of the fragmentation flags, as specified in Section [4.1.6](#).



- Notes

The number of octets should correspond to a typical data length transmitted by a Host, here set to 1000.

4.3.5.3 LL/CON/MAS/BV-04-C [Master Receiving Data]

- Test Purpose

Test that a master IUT is able to receive data from a slave device.

The Lower Tester acts in the slave role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



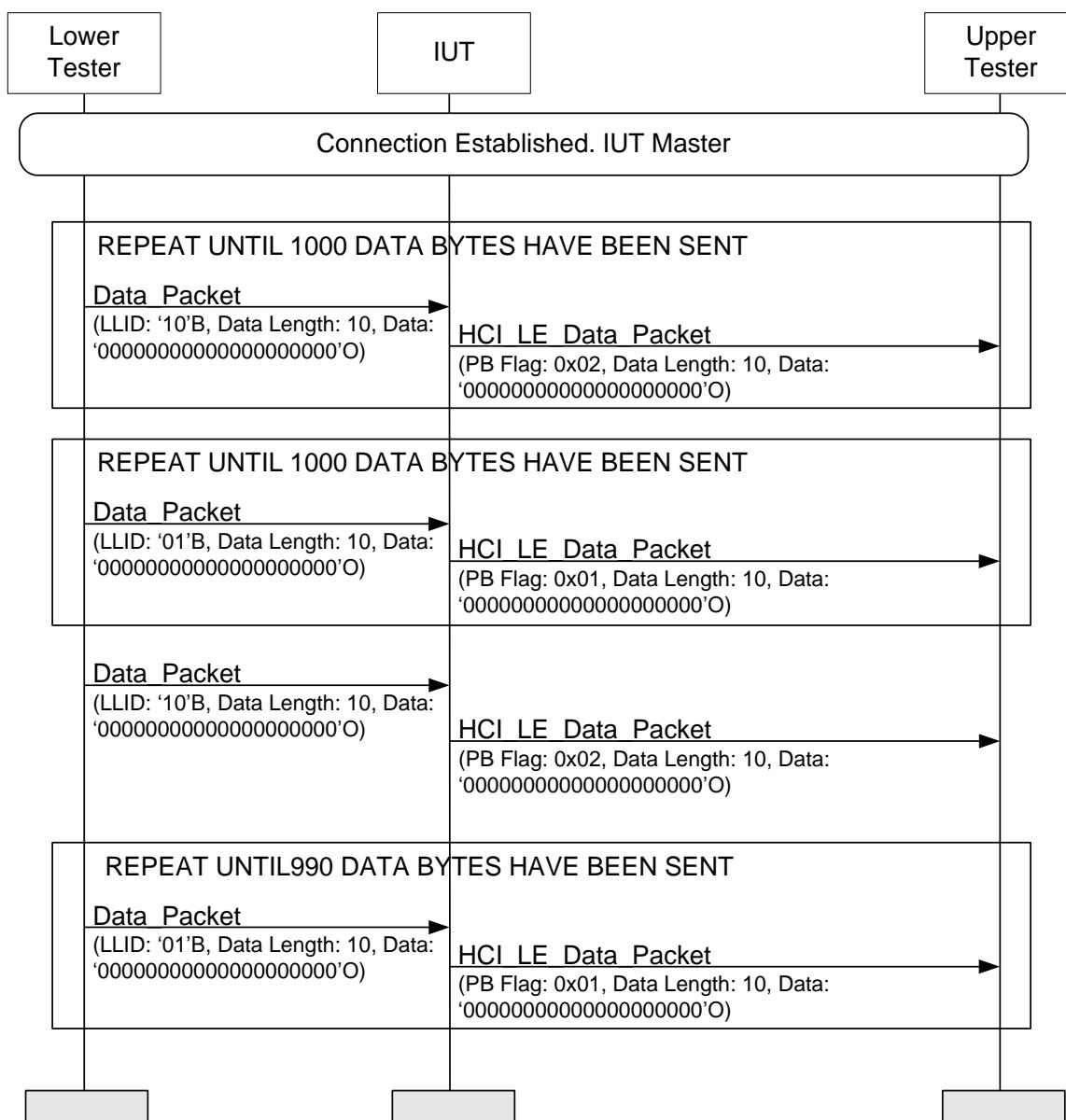


Figure 4.215: LL/CON/MAS/BV-04-C [Master Receiving Data]

- Configure Lower Tester to send 1000 data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
- Lower Tester expects a DATA packet from the IUT and sends a DATA packet in response using the acknowledgement scheme to the IUT on the same data channel. Repeat until the selected number of octets (1000) is acknowledged by the IUT.
- Upper Tester expects an HCI LE Data Packet event from the IUT containing a data element sent in step 2 and with the Packet_Boundary_Flag flag set to 0x02. Repeat until the selected number of octets (1000) is received in HCI LE Data Packet event.
- Configure Lower Tester to send 1000 data packets with the LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.
- Repeat step 2.
- Upper Tester expects HCI LE Data Packets from the IUT containing the data sent in step 5 and with the Packet_Boundary_Flag flag set to 0x01. Repeat until the selected number of octets (1000) is received in HCI LE Data Packet event.



7. Configure Lower Tester to send one data packet with LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00. After, send 990 data packets with LLID field set to 0x01 and using a payload length of 10 with the payload octets set to 0x00.
 8. Repeat step 2.
 9. Upper Tester expects HCI_LE_Data_Packets from the IUT containing the data sent in step 8. Repeat until the selected number of octets (1000) is received in HCI_LE_Data_Packet event. First HCI_LE_Data_Packet has the Packet_Boundary_Flag flag set to 0x02 and rest of HCI_LE_Data_Packet has the Packet_Boundary_Flag flag set to 0x01.
 10. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble steps).
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent,

The IUT reports the data correctly using HCI_Data_Packet events and the HCI fragmentation flags, as specified in Section 4.1.6.

4.3.5.4 LL/CON/MAS/BV-05-C [Master Sending and Receiving Data]

- Test Purpose

Test that a master IUT is able to send and receive data to/from a slave device.

The Lower Tester acts in the slave role in the connection, both submits data from the host of the IUT to transmit and sends data to the IUT according to the acknowledgement scheme and observes the data received and reported to the host of the IUT.

- Reference

[3] 4.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



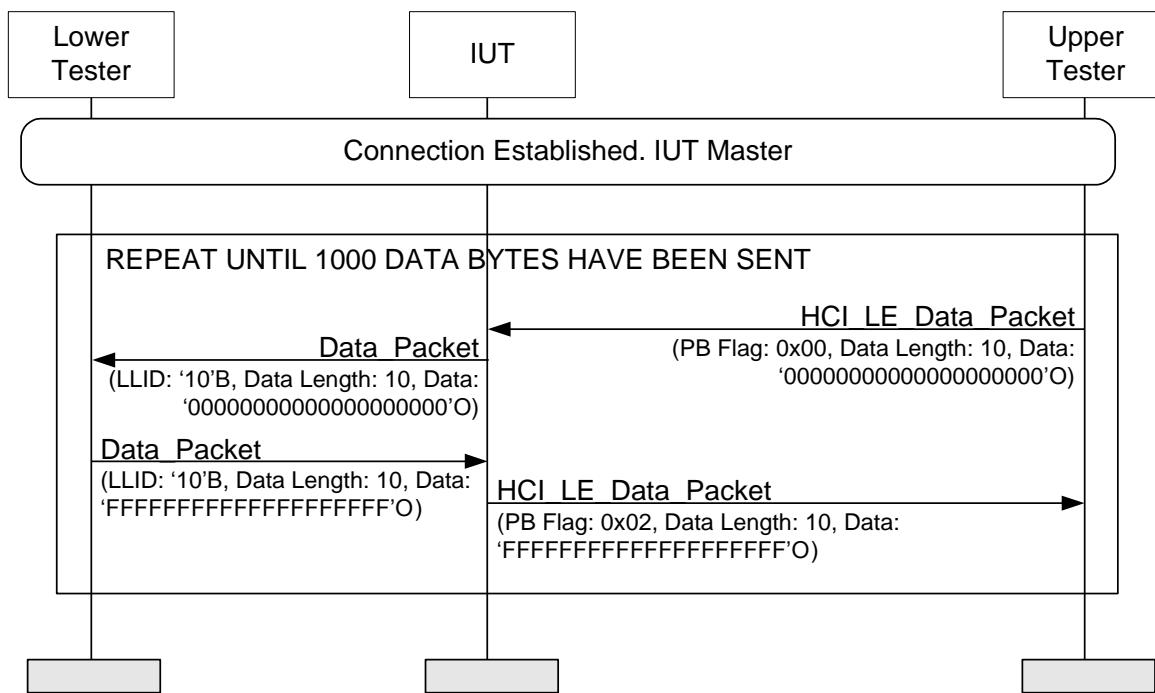


Figure 4.216: LL/CON/MAS/BV-05-C [Master Sending and Receiving Data]

1. Configure Lower Tester to send 100 data packets with the LLID field set to 0x02 in the first packet and to 0x01 in the following and using a payload length of 10 with the payload octets set to 0xFF.
 2. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, the data elements with the value 0x00, for a data total length of 10, with Packet_Boundary_Flag flag set to 0x00 in the first packet and set 0x01 in the following until the selected number of octets (1000) are successfully submitted.
 3. Expect an HCI_Number.Of_Completed_Packets event from the IUT using the connection handle, indicating a number of packets completed.
 4. Expect a DATA packet from the IUT, containing a data element submitted in 1: with the LLID field set to 0x02 in the first packet with data and 0x01 in the following.
 5. Respond with a DATA packet using the acknowledgement scheme to the IUT on the same data channel.
 6. Repeat steps 4-5 until all data sent in step 2 have been reported.
 7. Upper Tester expects HCI_LE_Data_Packets from the IUT containing the data sent in step 5 with the Packet_Boundary_Flag flag set to 0x02 in the first packet and set to 0x01 in the following. Payload octets shall be 0xFF.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent and reporting all data received, as specified in Section 4.1.6.

4.3.5.5 LL/CON/MAS/BV-07-C [Requesting Parameter Update]

- Test Purpose

Test that a master IUT is able to perform the connection parameter update procedure.

The Lower Tester acts in the slave role maintaining a connection and submits HCI commands to start connection parameter update as the host of the IUT, and then observes the procedure carried out by the IUT.



- Reference

[3] 5.1.1

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

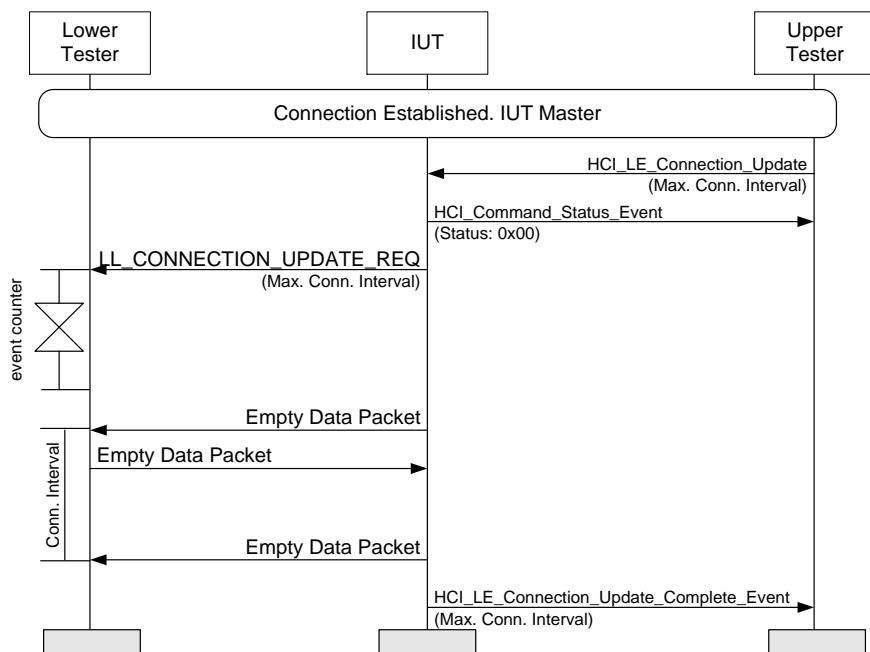


Figure 4.217: LL/CON/MAS/BV-07-C [Requesting parameter update]

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT, including parameters to use maximum connection interval and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects a CONNECTION_UPDATE_IND control packet from the IUT, with the parameters submitted in 1. Send an empty DATA packet in response using the acknowledgement scheme to the IUT on the same data channel. Repeat until the IUT has sent the packet following the connection update request, or up to a number of events (15):
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet:
4. Once the event count matches the time, the new parameters will be in use and the time between the data packet and the next empty packet sent by the IUT is one new connection interval (considering jitter).



5. Upper Tester expects an HCI_LE_Connection_Update_Complete event after the time of the update from the IUT containing the new connection parameters.
- Expected Outcome
Pass Verdict
The IUT transmits the request to update connection parameters.
The IUT maintains the connection with the new parameters in use.
The IUT reports the new parameters with an HCI event.

4.3.5.6 LL/CON/MAS/BV-08-C [Master Sending Termination]

- Test Purpose
Test that a master IUT is able to terminate a connection by sending the termination packet.
The Lower Tester acts in the slave role in the connection, on receiving the termination packet from the IUT, acknowledges it and observes the master stopping transmissions.
- Reference
[3] 5.1.6
- Initial Condition
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)
- Test Procedure

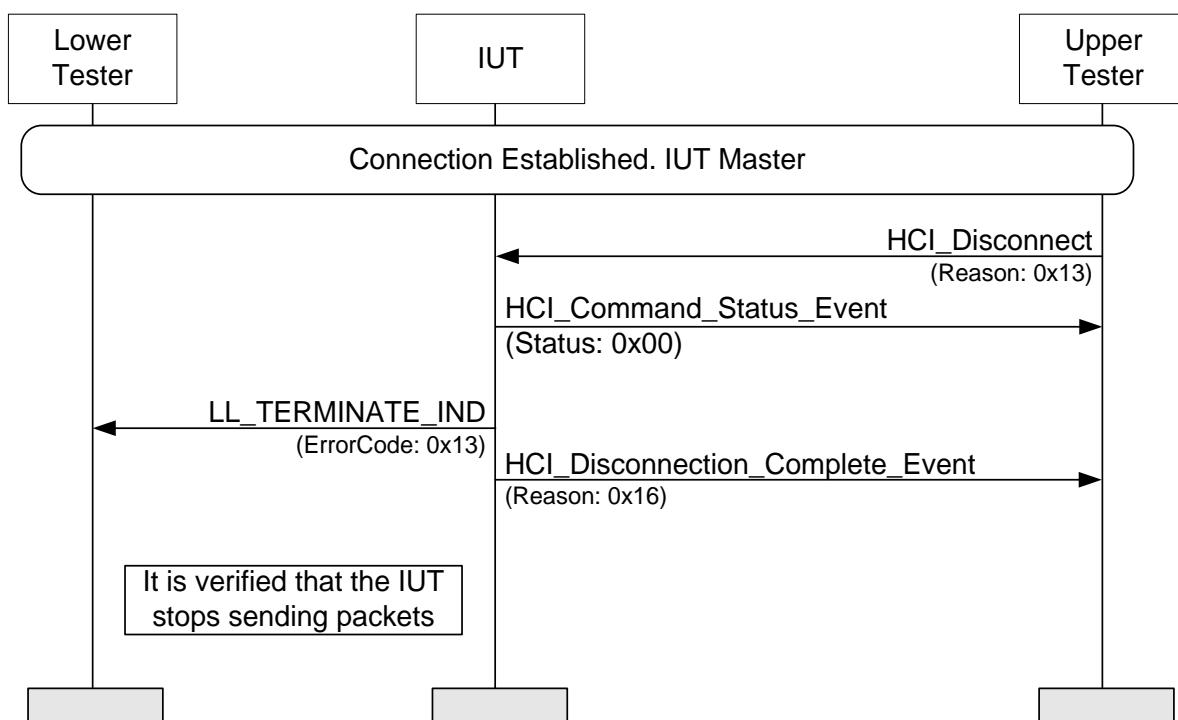


Figure 4.218: LL/CON/MAS/BV-08-C [Master Sending Termination]

1. Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status event in response.



2. Lower Tester expects the IUT to transmit a TERMINATE_IND packet. Lower Tester acknowledges the termination packet with an empty DATA response packet in the same event.
3. Lower Tester expects no master transmissions from the IUT up to a time equal to the connection supervision timeout:
4. Interleave with step 3: Upper Tester expects an HCI_Disconnection_Complete event from the IUT indicating that the connection termination procedure requested in Step 1 was successful.

- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT sending termination packets until acknowledgement from Lower Tester.

The IUT stops the master transmissions after the termination packets have been acknowledged.

The IUT reports the connection termination with an HCI event.

4.3.5.7 LL/CON/MAS/BV-09-C [Master Accepting Termination]

- Test Purpose

Test that a master IUT accepts connection termination by a slave sending the termination packet.

The Lower Tester acts in the slave role in the connection, sends the termination packet to the IUT and observes the master acknowledgement and stopping transmissions.

- Reference

[3] 5.1.6

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

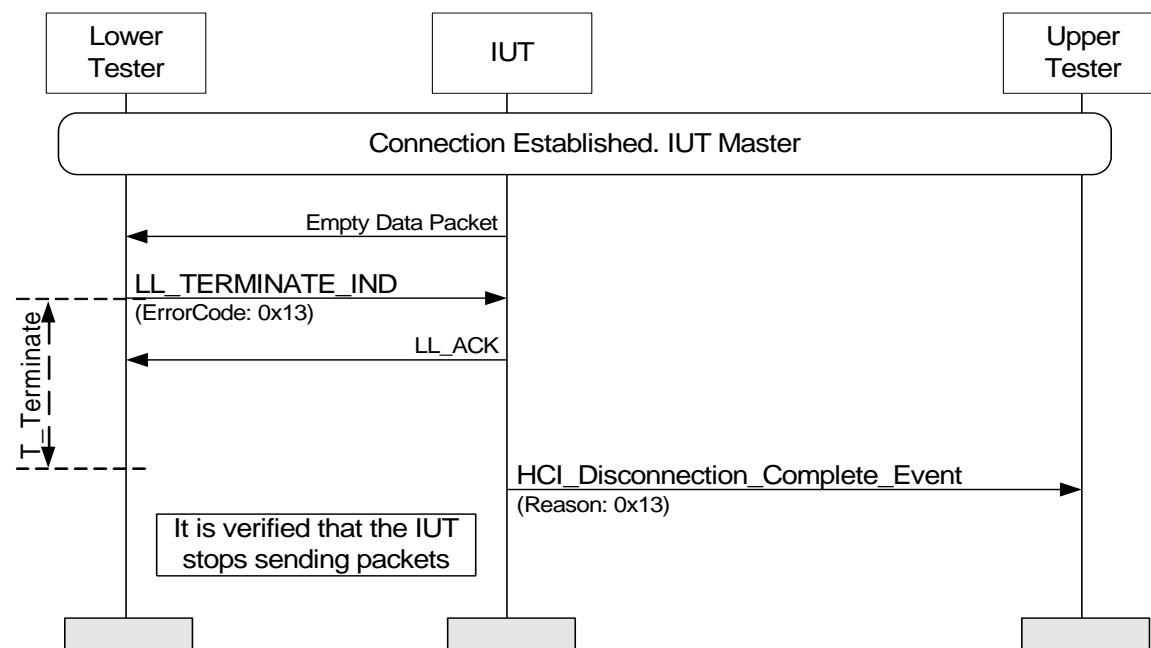


Figure 4.219: LL/CON/MAS/BV-09-C [Master Accepting Termination]



1. Lower Tester expects an empty DATA packet and transmits a TERMINATE_IND packet to the IUT in response. Expect an acknowledgement from the IUT in the following event.
 2. Lower Tester expects no master transmissions from the IUT up to a time equal to the connection supervision timeout:
 3. Interleave with step 2: Upper Tester expects an HCI_Disconnection_Complete event from the IUT indicating termination requested by the peer device and containing the connection handle from the preamble.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT acknowledging the termination packet before T_Terminate timer expires.

The IUT stops the master transmissions after it has acknowledged the TERMINATE_IND packet.

The IUT reports the connection termination with an HCI event.

4.3.5.8 LL/CON/MAS/BV-10-C [Master Supervision Timer]

- Test Purpose

Test that a master IUT terminates a connection by the supervision timer.

The Lower Tester acts in the slave role in the connection, stops responding in events, and then observes the IUT stopping transmissions and reporting the connection termination.

- Reference

[3] 4.5.2

- Initial Condition

Parameters: LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX,
LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, 0 slave latency, selected timeout, common channel map, not encrypted)



- Test Procedure

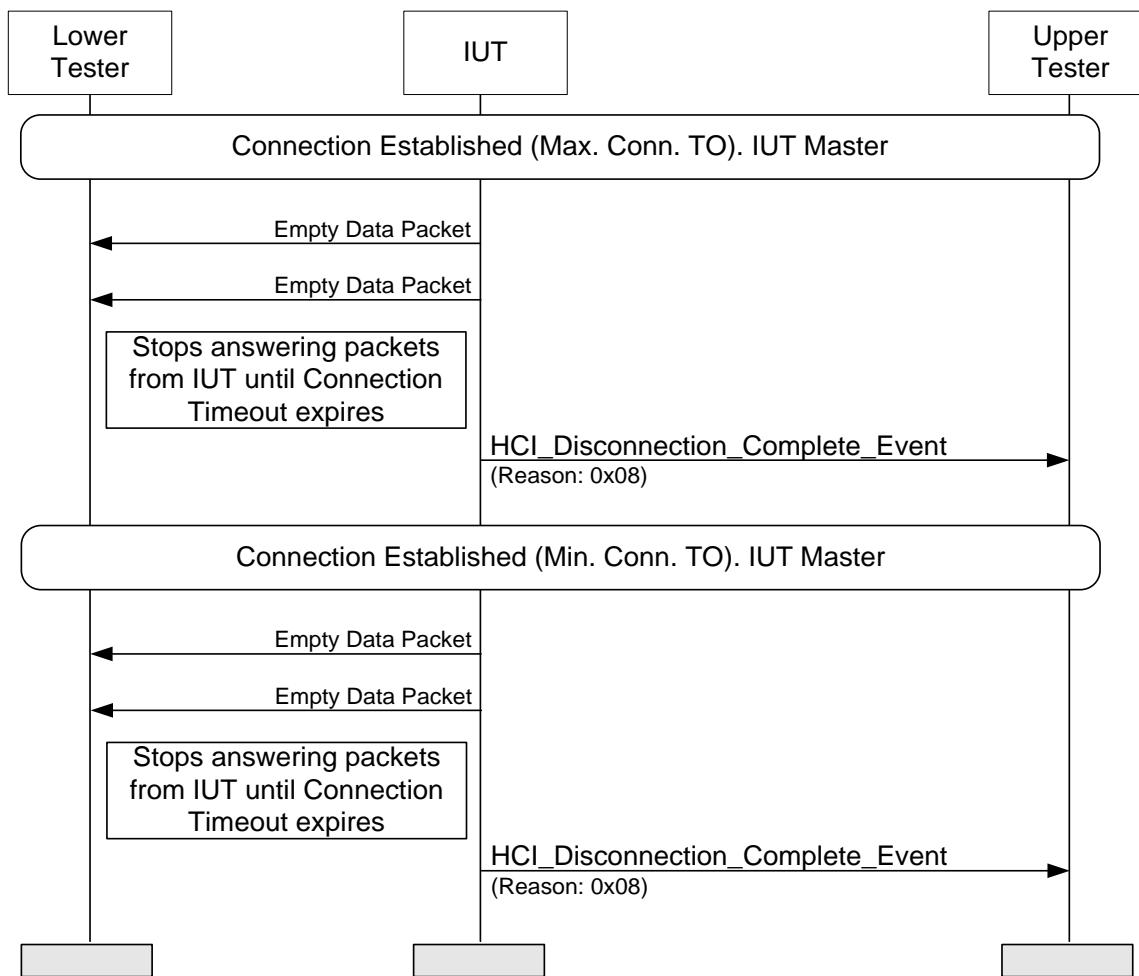


Figure 4.220: LL/CON/MAS/BV-10-C [Master Supervision Timer]

- The connection has been established with the maximum connection supervision timeout and connection interval values supported and use zero slave latency.
 - Lower Tester stops responding to the master transmissions after a specific event and until connection timeout expires.
 - Upper Tester expects an HCI_Disconnection_Complete event from the IUT, containing the connection handle and termination by timeout as the reason after a time no shorter than the connection supervision timeout value from the last event responded in.
 - Establish the connection with the minimum connection supervision timeout and connection interval values supported and use zero slave latency.
 - Repeat steps 2-3.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT reporting the connection termination event after the selected connSupervisionTimeout value.



4.3.5.9 LL/CON/MAS/BV-13-C [Feature Setup Request]

- Test Purpose

Test that a connected master IUT requests and performs the feature setup procedure, activating the correct features.

The Lower Tester acts in the slave role in a maintained connection and responds to the request from the IUT to perform feature setup.

- Reference

[3] 4.6, 5.1.4

- Initial Condition

Parameters: Parameters: LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, selected connection interval, selected slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure

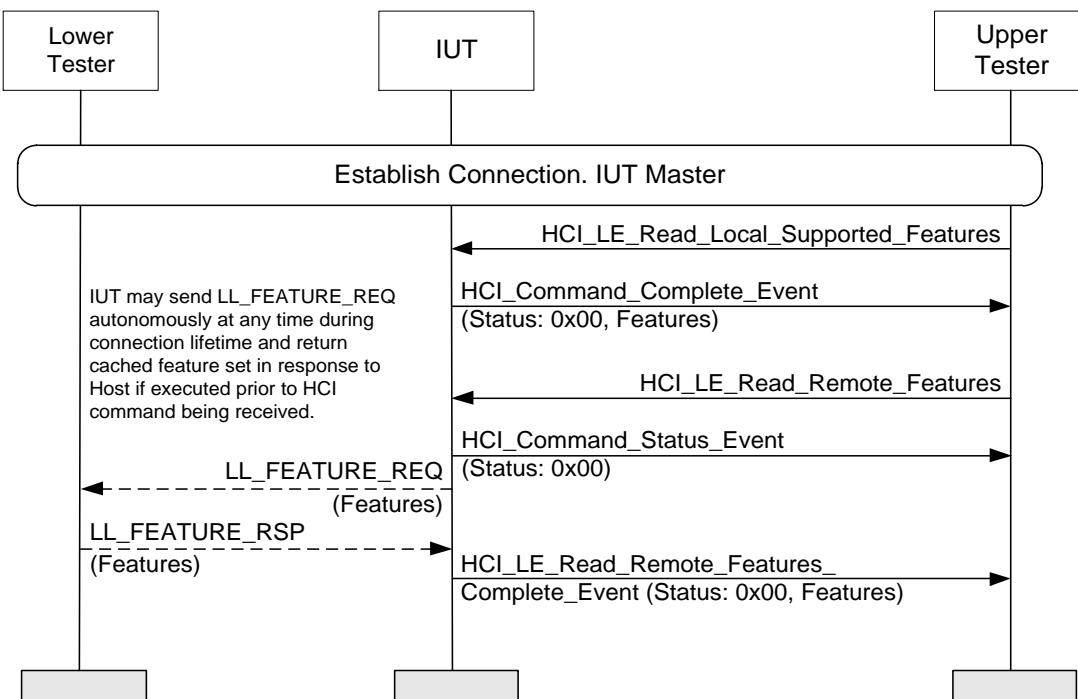


Figure 4.221: LL/CON/MAS/BV-13-C [Feature Setup Request]

- Upper Tester sends an HCI_LE_Read_Local_Supported_Features command and receives an HCI_Command_Complete_Event with the correct feature set value.
- Upper Tester sends an HCI_LE_Read_Remote_Features command.
- The IUT sends an LL_FEATURE_REQ PDU including the configured feature set or may have autonomously sent an LL_FEATURE_REQ prior to receiving the HCI command from the Upper Tester. In either case the IUT waits for an LL_FEATURE_RSP PDU from the Lower Tester.
- The IUT sends the HCI_LE_Read_Remote_Features_Complete event to the Upper Tester.



- Expected Outcome

Pass Verdict

All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT requesting the feature information and acknowledging the reply,

The IUT reports the feature setup procedure completed with an HCI event containing the correct used features for the connection.

- Notes

The command called LE Read Remote Features and the event called LE Read Remote Features Complete in this test and the 5.0 Core Specification were called LE Read Remote Used Features and LE Read Remote Used Features Complete in the 4.0, 4.1, and 4.2 Core Specifications.

4.3.5.10 LL/CON/MAS/BV-14-C [Master Retransmission Request]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme.

The Lower Tester acts in the slave role, using invalid checksums to prompt a repeated retransmission request from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout of 900 ms.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.



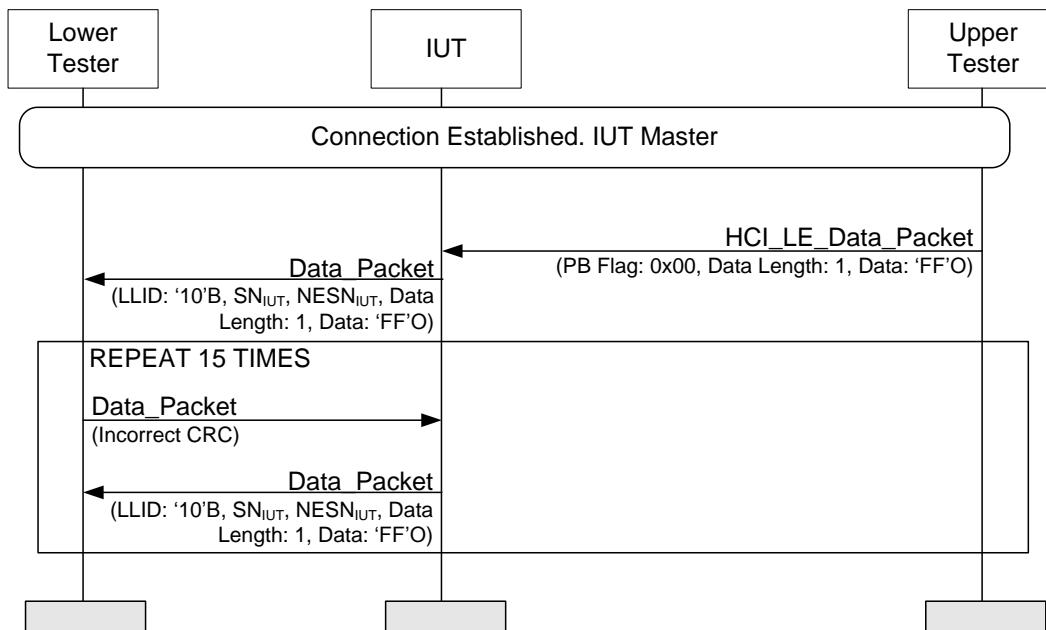


Figure 4.222: LL/CON/MAS/BV-14-C [Master Retransmission Request]

- Configure Upper Tester to submit data elements to the IUT with the HCI LE Data Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.
 - Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters and sends a LL DATA packet with an invalid checksum in response to every packet, TIFS after the packet from the IUT. Repeat 15 times.
 - During the repetition procedure, observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the next SN and by using the next NESN where the NESN matches the current SN in the previous packet correctly received.
 - Lower Tester expects the IUT to indicate in the following packets received correctly that the previous packet was not received correctly, with the SN matching the current NESN and the NESN matching the next SN.
 - Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The IUT transmits negative acknowledgements when sent a packet with an invalid CRC.

4.3.5.11 LL/CON/MAS/BV-15-C [Master Retransmission]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme and retransmit a data packet on a negative acknowledgement.

The Lower Tester acts in the slave role, using negative acknowledgements prompts repeated retransmission requests from the IUT.

- Reference

[3] 4.5.9



- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted).

- Test Procedure

- Execute the test procedure using the common connection interval and slave latency.
- Use a connection supervision timeout of 900 ms.
- Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.

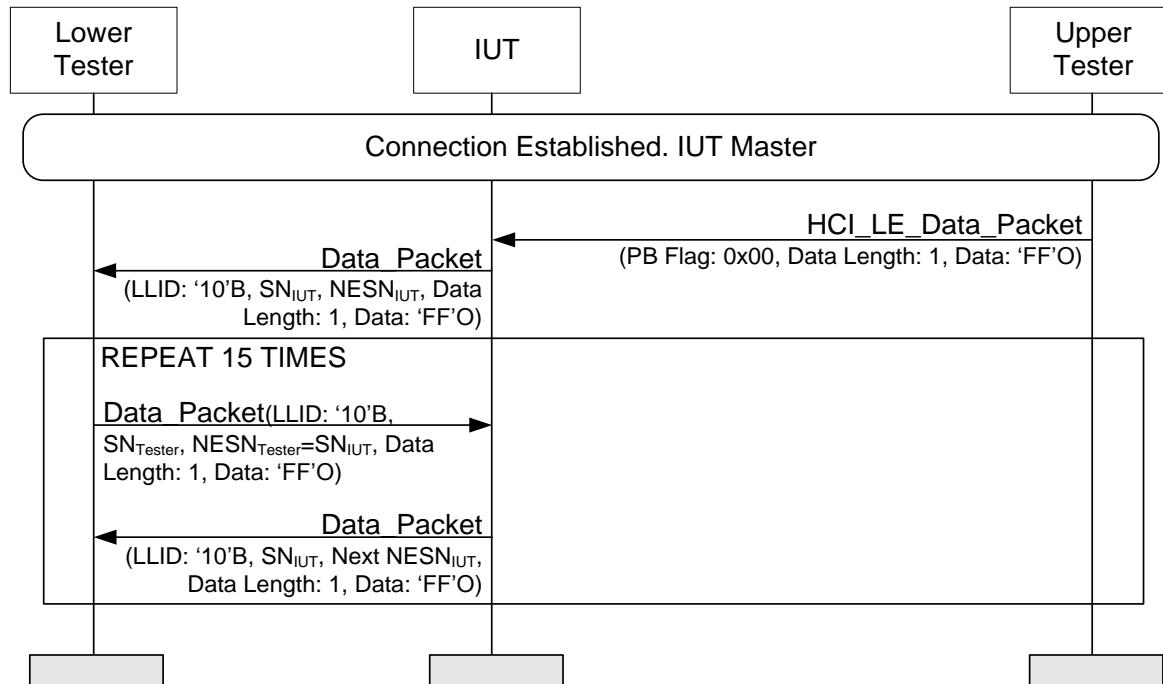


Figure 4.223: LL/CON/MAS/BV-15-C [Master Retransmission]

- Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.
- Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA in response to every packet, TIFS after the packet received from the IUT, not acknowledge the packet received by using the current NESN. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the current SN
- Lower Tester expects the IUT to retransmit the packet in the following packets received correctly, with the SN matching the current NESN and the NESN matching the next SN.
- Repeat steps 2-3 15 times.
- Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).

- Expected Outcome

Pass Verdict

The IUT retransmits the same payload when asked for a retransmission.



4.3.5.12 LL/CON/MAS/BV-16-C [Master Acknowledgement Repetition]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme and repeats a positive acknowledgement of a packet.

The Lower Tester acts in the slave role, using negative acknowledgements to prompt a retransmission of an acknowledgement from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted).

- Test Procedure

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout of 900 ms.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.

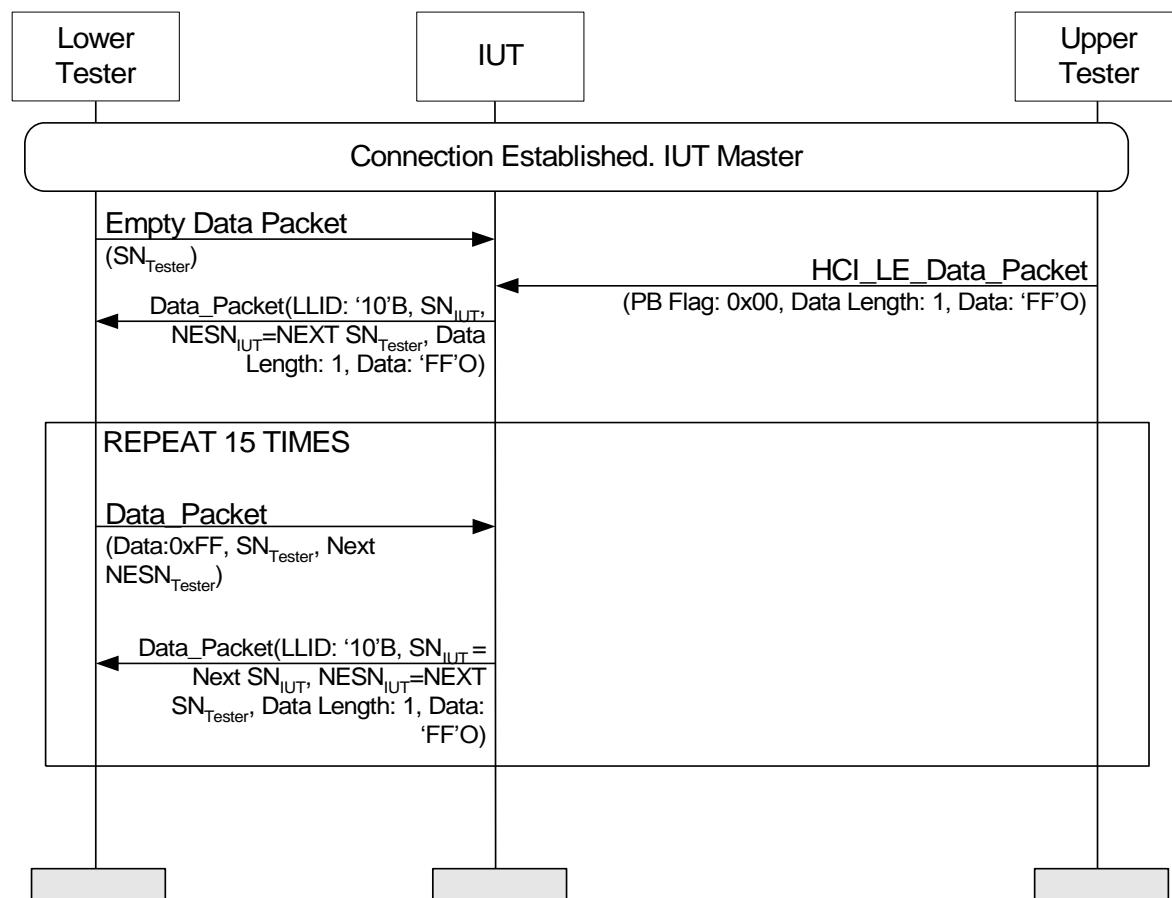


Figure 4.224: LL/CON/MAS/BV-16-C [Master Acknowledgement Repetition]



1. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data elements with the value 0xFF with Packet_Boundary_Flag flag set.
 2. Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet in response to the IUT to every packet, TIFS after the packet received from the IUT. Not to recognize an acknowledgement, use the current SN where in the previous packet received the NESN is the next SN, but acknowledge packets correctly with the next NESN where the current NESN matches the SN in the previous packet correctly received.
 3. Lower Tester expects the IUT to repeat the acknowledgement in the following packets received correctly, with the SN matching the current NESN and the NESN matching the next SN.
 4. Repeat steps 2-3 15 times.
 5. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The IUT retransmits acknowledgements when negative acknowledgements are received.

4.3.5.13 LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme and preserve the packet sequence numbering in the case of a lost negative acknowledgement.

The Lower Tester acts in the slave role, moving to the next packet after a negative acknowledgement to prompt a repeated negative acknowledgement from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout of 900 ms.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.



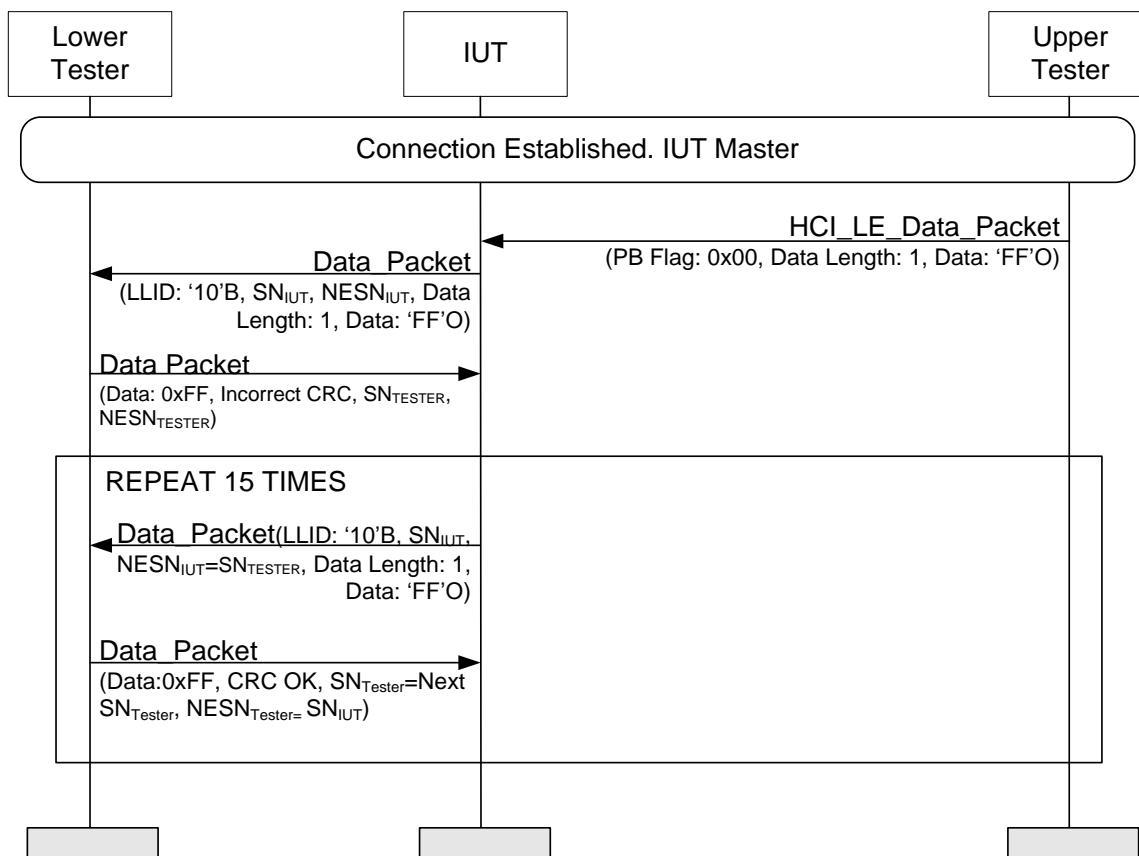


Figure 4.225: LL/CON/MAS/BV-17-C [Master Lost Negative Acknowledgement]

- Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data element with the value 0xFF with Packet_Boundary_Flag flag set.
 - Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet with a varying checksum to every packet, TIFS after the packet from the IUT. To prompt the IUT to produce a negative acknowledgement, use the next SN for one packet to send with an invalid checksum, where the NESN in the previous packet received correctly is the next SN
 - Lower Tester expects the IUT with negative acknowledgement and sends a DATA packet with a valid checksum but use the SN equal to the SN in the previous Lower Tester packet. Acknowledge packets by using the next NESN where the current NESN matches the SN in the previous packet correctly received.
 - Lower Tester expects the IUT to continue the negative acknowledgement in the following packets received correctly, with the NESN matching the current SN.
 - Repeat step 3-4 15 times.
 - Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The IUT retransmits a negative acknowledgement when being transmitted the next payload.



4.3.5.14 LL/CON/MAS/BV-18-C [Master Latency Retransmission Request]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme with the slave using latency.

The Lower Tester acts in the slave role, using slave latency and invalid checksums and observe the master handling the latency and retransmission requests from the IUT.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure

Execute the test procedure using the common connection interval and slave latency.

Use a connection supervision timeout as defined by the TSPX_conn_timeout_max IXIT entry.

Use the common data channel selection parameters.

The test uses the common current and next variables for SN and NESN.



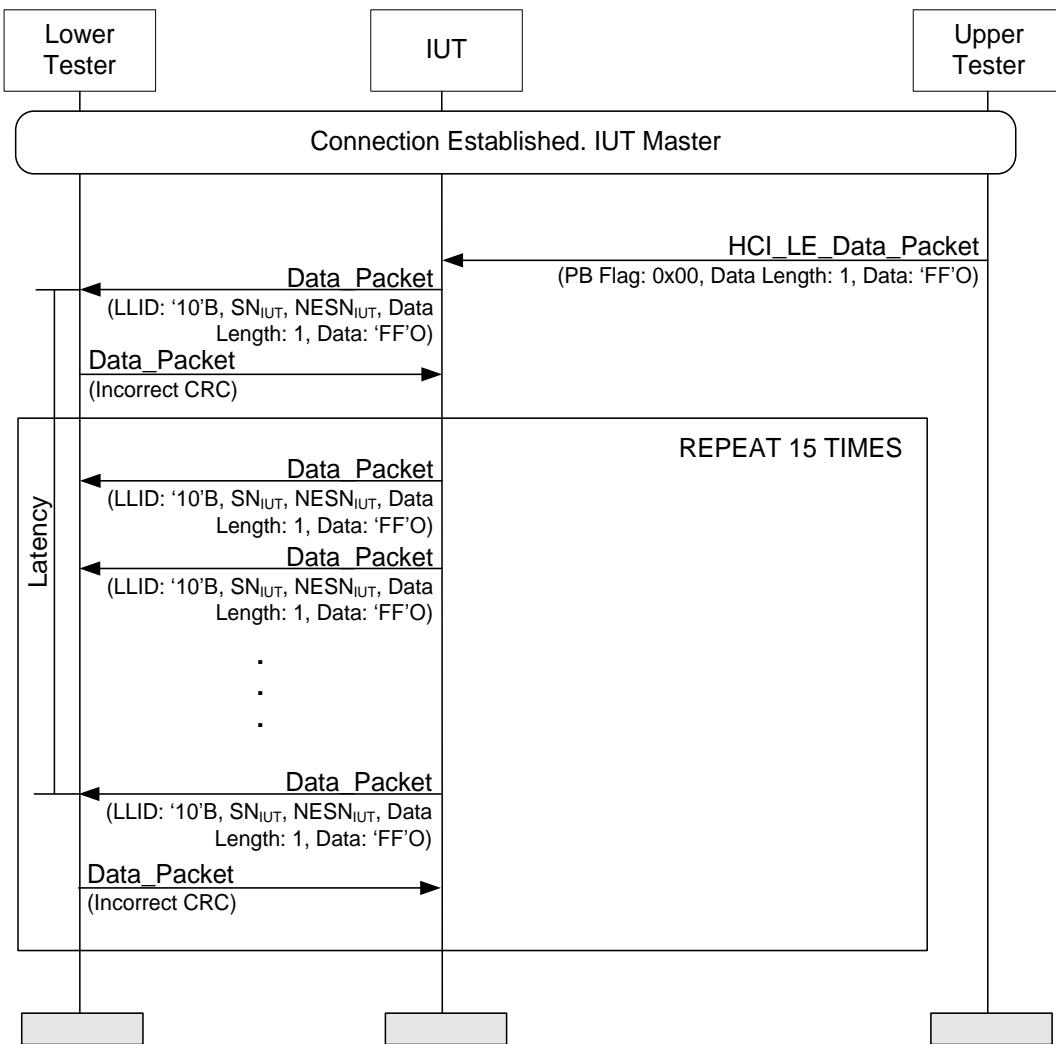


Figure 4.226: LL/CON/MAS/BV-18-C [Master Latency Retransmission Request]

- Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, one data element with the value 0xFF with Packet_Boundary_Flag flag set.
- Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Lower Tester sends a DATA packet with an invalid checksum only in events required by the slave latency parameter, TIFS after the packet received from the IUT. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the previous packet correctly received is the current SN and by using the next NESN where the current NESN matches the current SN in the previous packet correctly received.
- Lower Tester expects the IUT to indicate in the following packets received correctly that the previous packet was not received correctly or no response was received, with the SN matching the current NESN and the NESN matching the current SN.
- Repeat steps 2-3 15 times.
- Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).

- Expected Outcome

Pass Verdict

The IUT uses retransmissions to handle slave latency.

4.3.5.15 LL/CON/MAS/BV-19-C [Connection Control Timeout]

- Test Purpose

Test that a master IUT terminates the link layer connection if a transaction is not completed before the connection control transaction timer expires.

The Lower Tester acts in the slave role in the connection and avoids that a transaction is completed.

- Reference

[3] 5.2

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

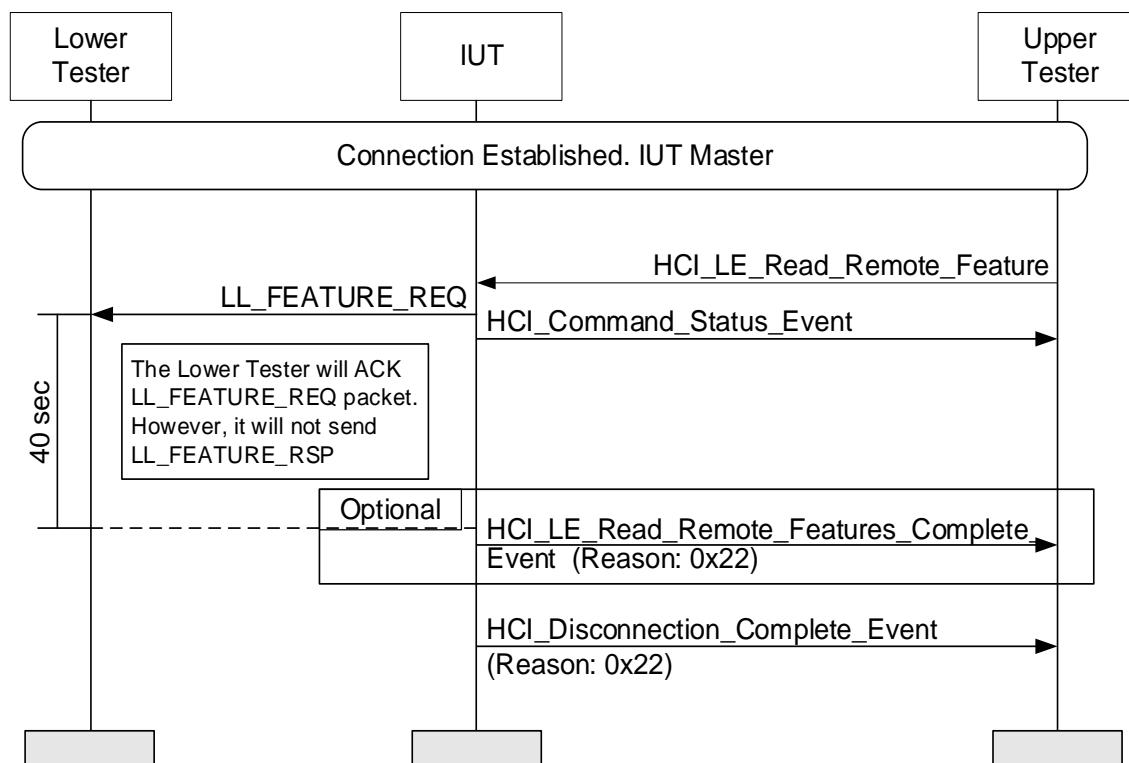


Figure 4.227: LL/CON/MAS/BV-19-C [Connection Control Timeout]

1. Upper Tester sends an HCI_LE_Read_Remote_Feature command and receives an HCI_Command_Status_Event.
2. The IUT sends an LL_FEATURE_REQ PDU.
3. Lower Tester acknowledges the LL_FEATURE_REQ PDU but does not send an LL_FEATURE_RSP PDU.



4. IUT sends the event HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester.

- Expected Outcome

Pass Verdict

The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires.

4.3.5.16 LL/CON/MAS/BV-20-C [Master Request Version]

- Test Purpose

Test that a connected master IUT requests and performs the Version Exchange procedure.

The Lower Tester acts in the slave role in a maintained connection and responds to the request from the IUT to perform version exchange.

- Reference

[3] 5.1.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

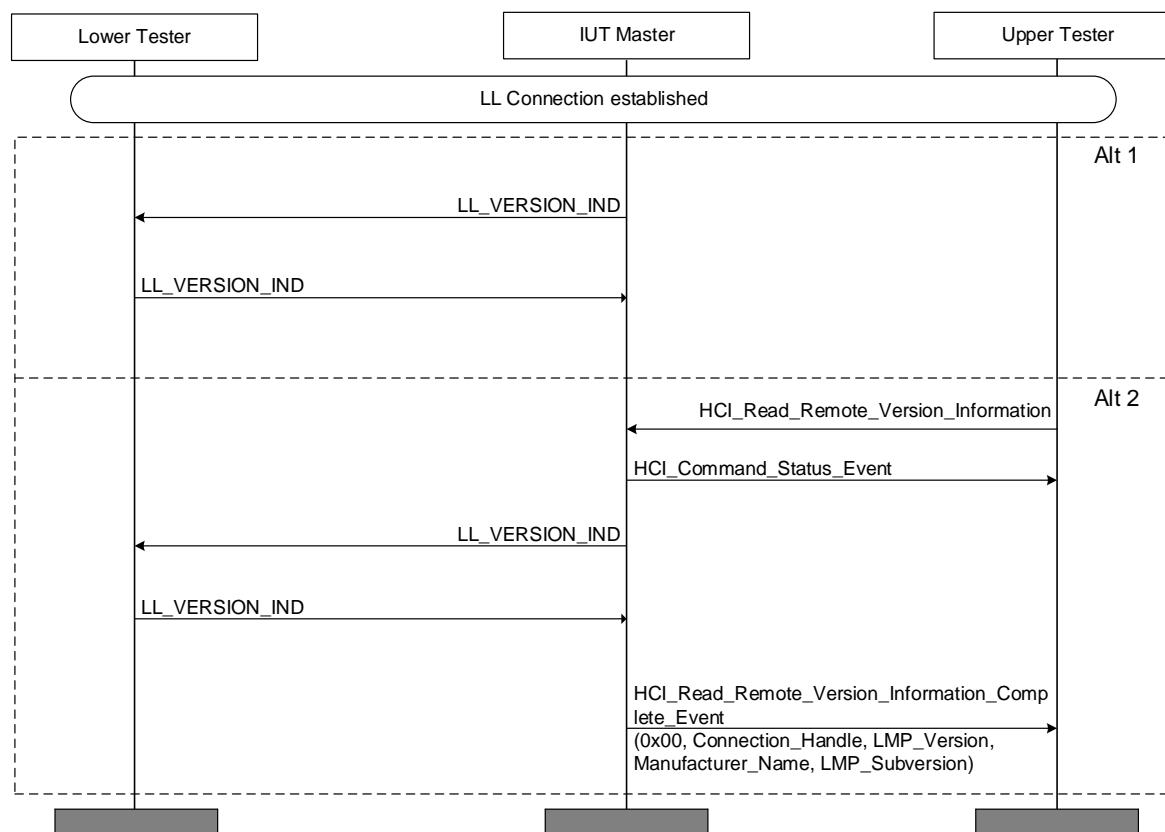


Figure 4.228: LL/CON/MAS/BV-20-C [Master Request Version]



1. Once connection is established, if the IUT has sent the Lower Tester a Version_Ind packet containing IUT version information by itself, the Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
 2. If the IUT does not send the Lower Tester a Version_Ind packet containing IUT version information by itself, the Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT and receives an HCI_Command_Status_Event as a response.
 3. The IUT sends a Version_Ind packet containing IUT version information.
 4. The Lower Tester acknowledges the Version_Ind packet and sends a Version_Ind packet containing Tester version information.
 5. The IUT sends the event HCI_Read_Remote_Version_Information_Complete_Event with Tester version information to the Upper Tester.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT requesting the version information and acknowledging the reply.

If the procedure was initiated by the Upper Tester as described in Step 2, the IUT reports the version requested completed with an HCI event.

4.3.5.17 LL/CON/MAS/BV-21-C [Master Respond Version]

- Test Purpose

Test that a connected master IUT responds to the request from the Lower Tester to perform the version exchange procedure.

The Lower Tester acts in the slave role in a maintained connection and requests to perform version exchange.

- Reference

[3] 5.1.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

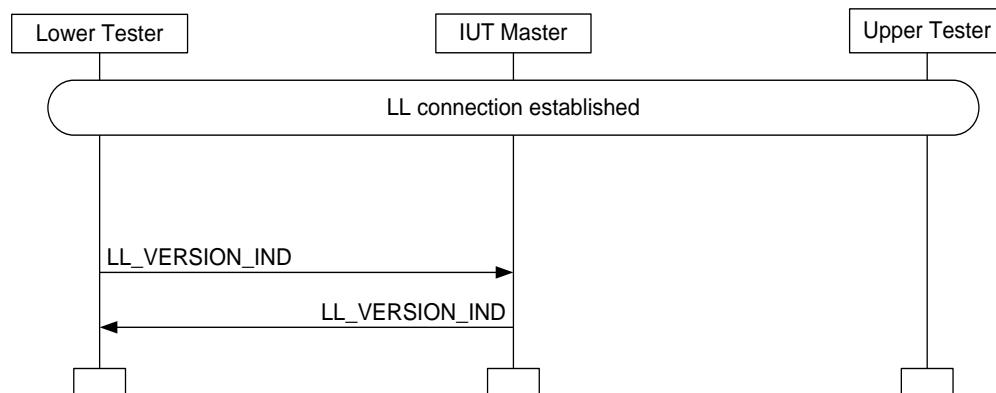


Figure 4.229: LL/CON/MAS/BV-21-C [Master Respond Version]



1. Configure the Lower Tester to send Version_Ind packet
 2. Lower Tester sends a Version_Ind packet containing Tester version information.
 3. The IUT acknowledges the Version_Ind packet and sends a Version_Ind packet containing IUT version information.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT responding to the version information.

Inconclusive Verdict

The IUT sends an LL_VERSION_IND PDU before the Lower Tester does.

4.3.5.18 LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]

- Test Purpose

Test that a master IUT is able to maintain a connection using the acknowledgement scheme.

The Lower Tester acts in the slave role, maintaining the connection and checking that the IUT uses correctly the acknowledgement scheme.

- Reference

[3] 4.5.9

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, selected timeout, common channel map, not encrypted)

- Test Procedure

Execute the test procedure using the common connection interval and slave latency. Use a connection supervision timeout of 900 ms. Use the common data channel selection parameters. The test uses the common current and next variables for SN and NESN.



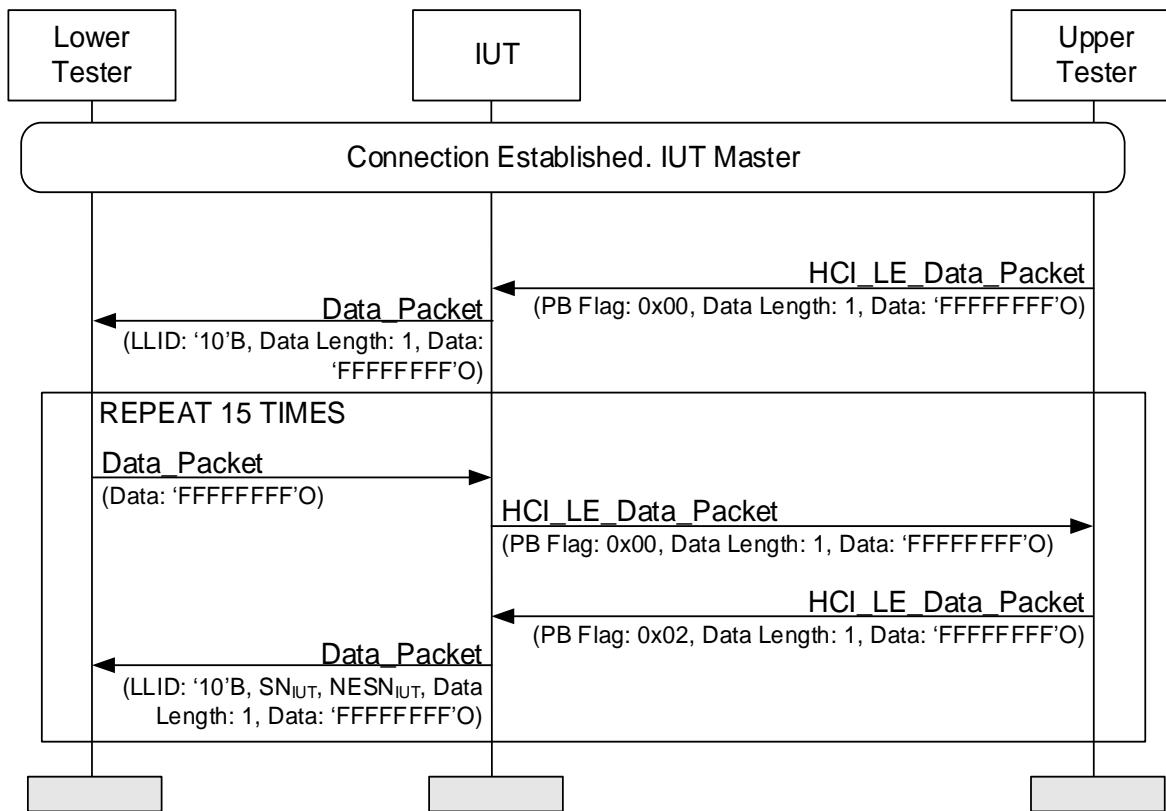


Figure 4.230: LL/CON/MAS/BV-22-C [Master Acknowledgement Scheme]

- Configure Upper Tester to submit one data element to the IUT with the **HCI LE Data Packet** command using the connection handle, the data elements with the value 0xFFFFFFFF, for a data total length of 4, with **Packet_Boundary_Flag** flag set.
 - Configure Lower Tester to send one data packets with a payload length of 4 with the payload octet set to 0xFFFFFFFF.
 - Lower Tester expects a DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters. Expect the SN in a correctly received packet to match the current NESN and the NESN to match the next SN.
 - Lower Tester sends a DATA packet in response to every packet, T_IFS after the packet from the IUT. Observe the acknowledgement scheme by using the next SN for every packet to send where the NESN in the packet correctly received is the next SN and by using the next NESN where the current NESN matches the SN in the packet correctly received.
 - The IUT sends the event **HCI LE Data Packet** to the Upper Tester including the data sent by Lower Tester.
 - Repeat steps 1-5 15 times.
 - Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT using the normal acknowledgement scheme operation.



4.3.5.19 LL/CON/MAS/BV-23-C [Responding to Feature Exchange]

- Test Purpose

Test that a connected master IUT responds to the feature exchange procedure and activates the correct features when requested.

The Lower Tester acts in the slave role in a maintained connection, transmits the request to perform feature exchange and observes the IUT responding.

- Reference

[8] 5.1.4.2

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

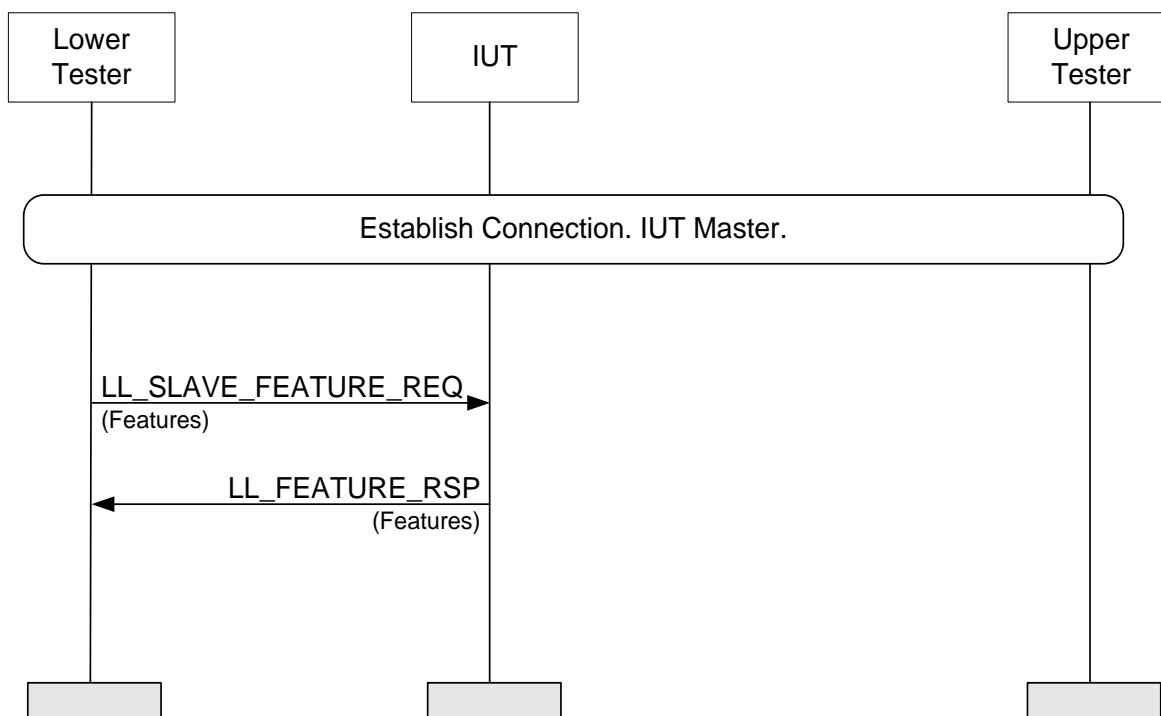


Figure 4.231: LL/CON/MAS/BV-23-C [Responding to Feature Exchange]

1. Lower Tester sends an LL_SLAVE_FEATURE_REQ PDU including the Lower Tester's feature set and waits for an LL_FEATURE_RSP PDU.
2. The IUT responds to the LL_SLAVE_FEATURE_REQ PDU with an LL_FEATURE_RSP PDU.



- Expected Outcome

Pass Verdict

All bits in the feature set marked as Masked to Peer received by the Lower Tester are cleared.

The test procedure is executed successfully, with the IUT responding with the feature response.

4.3.5.20 LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device accepts the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and accepts the IUT's request. The actual parameters used by the IUT may be different from the parameters provided by the Upper Tester.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

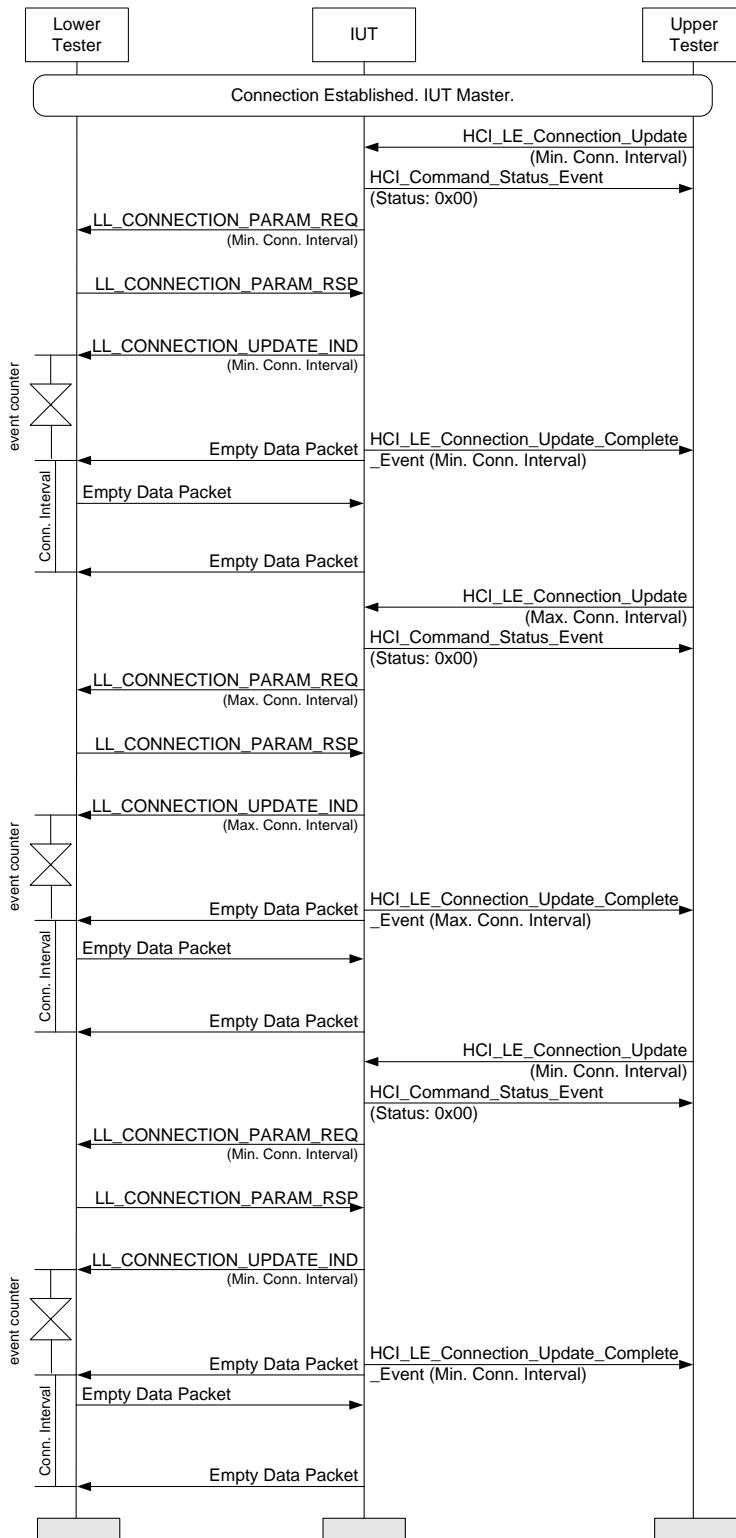


Figure 4.232: LL/CON/MAS/BV-24-C [Initiating Connection Parameter Request – Accept]

Case 1:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester expects a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the maximum connection interval, no latency and maximum connection supervision timeout and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester expects a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester accepts the IUT's request and responds with an LL_CONNECTION_PARAM_RSP PDU. Lower Tester expects a packet from the IUT acknowledging the connection parameter response followed by an LL_CONNECTION_UPDATE_IND.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
5. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.



- Expected Outcome

Pass Verdict

- For all three cases described in the test procedure, the following conditions shall occur:
 - The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
 - The IUT maintains the connection with the new parameters selected by the IUT in use.
 - The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.21 LL/CON/MAS/BV-25-C [Initiating Connection Parameter Request – Reject]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device rejects the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT's request.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
 LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

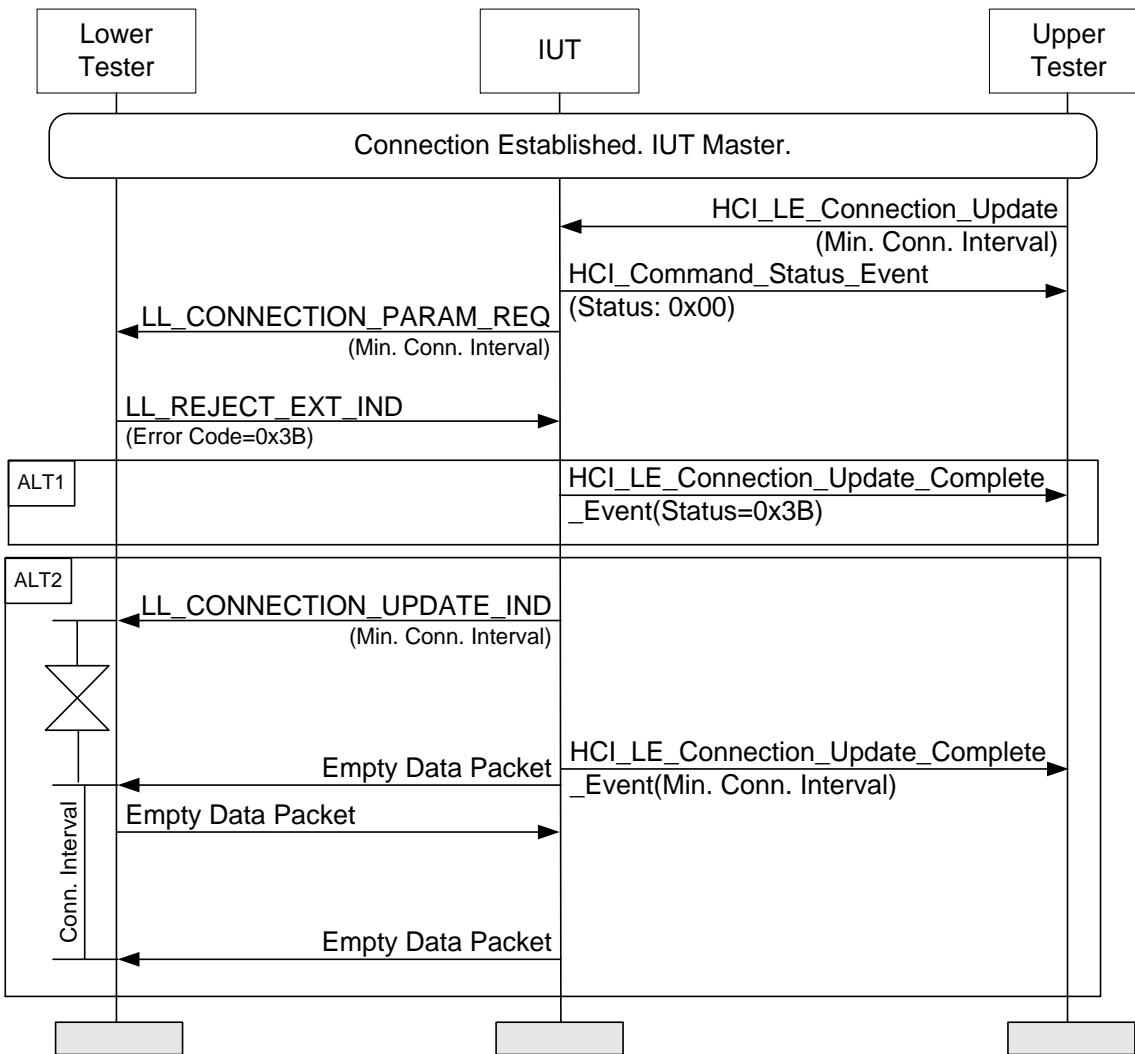


Figure 4.233: LL/CON/MAS/BV-25-C [Initiating Connection Parameter Request – Reject]

- Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
- Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester rejects the IUT's request by issuing an LL_REJECT_EXT_IND PDU containing ErrorCode 0x3B. Lower Tester expects a packet from the IUT acknowledging the LL_REJECT_EXT_IND PDU.

Alternative 1:

- Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the error code sent by the Lower Tester (0x3B).

Alternative 2:

- Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester).



- Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
- Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
- Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.

The IUT either reports an HCI_LE_Connection_Update_Complete event containing the error code sent by the Lower Tester (0x3B) or reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.22 LL/CON/MAS/BV-26-C [Initiating Connection Parameter Request – same procedure collision]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when there is a procedure collision between the IUT's connection parameter request and the remote device's connection parameter request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new connection parameter request procedure upon receiving the IUT's connection parameter request to cause a procedure collision. The test case expects the IUT to reject the slave's connection parameter request and then proceed with its own connection parameter request procedure.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
 LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

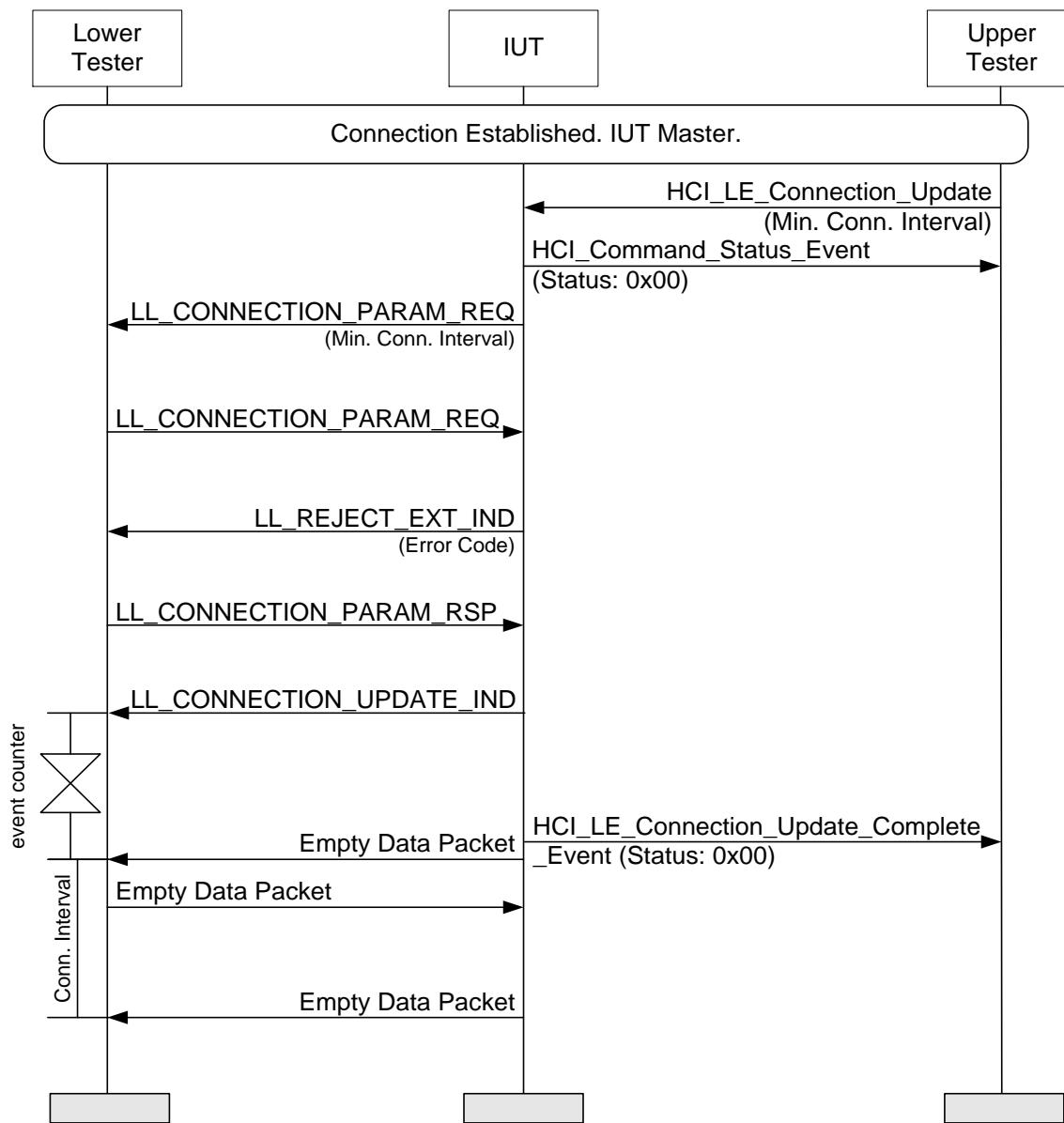


Figure 4.234: LL/CON/MAS/BV-26-C [Initiating Connection Parameter Request – same procedure collision]

- Upper Tester sends an **HCI LE_Connection_Update** command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an **HCI_Command_Status** event from the IUT in response.
- Lower Tester expects an **LL_CONNECTION_PARAM_REQ** control PDU from the IUT (the actual parameters in the **LL_CONNECTION_PARAM_REQ** PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an **LL_CONNECTION_PARAM_REQ**.
- Lower Tester expects the IUT to reject the Lower Tester's connection parameter request using an **LL_REJECT_EXT_IND** with reason code 0x23.
- Lower Tester responds to the IUT's connection parameter request using an **LL_CONNECTION_PARAM_RSP** and expects the IUT to respond with an **LL_CONNECTION_UPDATE_IND**.



5. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
 6. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
 7. At the time of the update start maintaining the connection with the new parameters.
 8. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and rejects the Lower Tester's request to update the connection parameters and then expects the connection parameter response from the Lower Tester.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.23 LL/CON/MAS/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

- Test Purpose

Test that a master IUT is able to perform the channel map update procedure when there is a procedure collision between the IUT's channel map update and the Lower Tester's connection parameter request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the channel map update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameter request procedure upon receiving the IUT's channel map update to cause a procedure collision. The test case expects the IUT to reject the slave's connection parameter request and then proceed with its own channel map update procedure.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

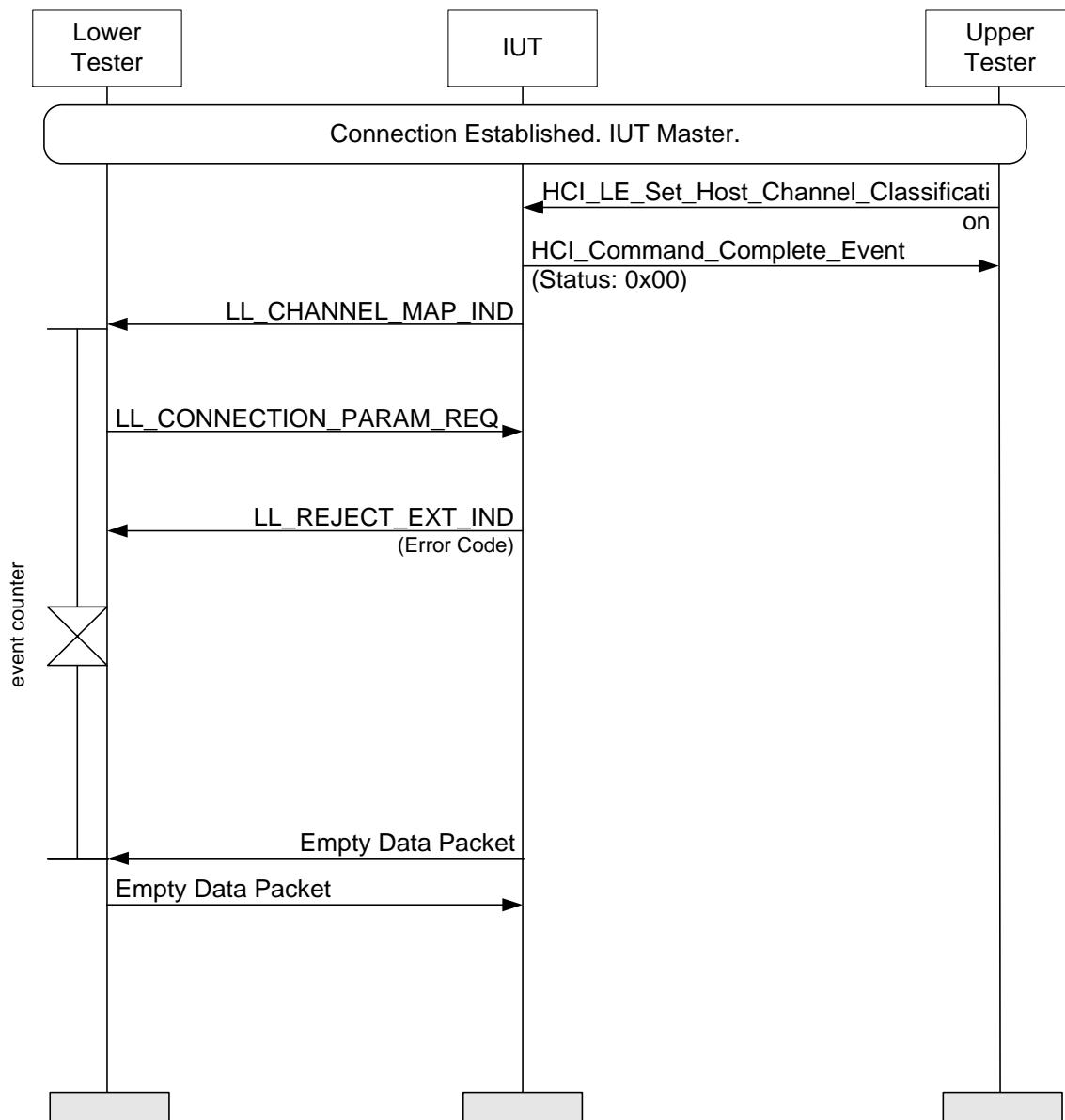


Figure 4.235: LL/CON/MAS/BV-27-C [Initiating Connection Parameter Request – different procedure collision – channel map update]

- Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT setting the channel map to only use even channels. Upper Tester expects an HCI_Command_Complete event from the IUT in response.
- Lower Tester expects an LL_CHANNEL_MAP_IND control PDU from the IUT, with the parameters submitted in step 1.
- Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT with connection parameters set to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) as an acknowledgement packet to the LL_CHANNEL_MAP_IND PDU.
- Lower Tester expects the IUT to reject the LL_CONNECTION_PARAM_REQ PDU using an LL_REJECT_EXT_IND with ErrorCode 0x2A.
- Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the channel map update request.

- Expected Outcome

Pass Verdict

The IUT transmits the LL_CHANNEL_MAP_IND PDU to update the channel map and rejects the Lower Tester's connection parameter request.

The IUT maintains the connection with the Lower Tester with the updated data channel selection parameters after the assigned event.

4.3.5.24 LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

- Test Purpose

Test that a master IUT is able to perform the encryption start procedure when there is a procedure collision between the IUT's encryption start procedure and the Lower Tester's connection parameter request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the encryption procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a connection parameter request procedure upon receiving the IUT's encryption start to cause a procedure collision. The test case expects the IUT and Lower Tester to complete the encryption start procedure followed by completing the connection parameter request procedure.

- Reference

[8] 5.1.7, 5.1.3.1

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

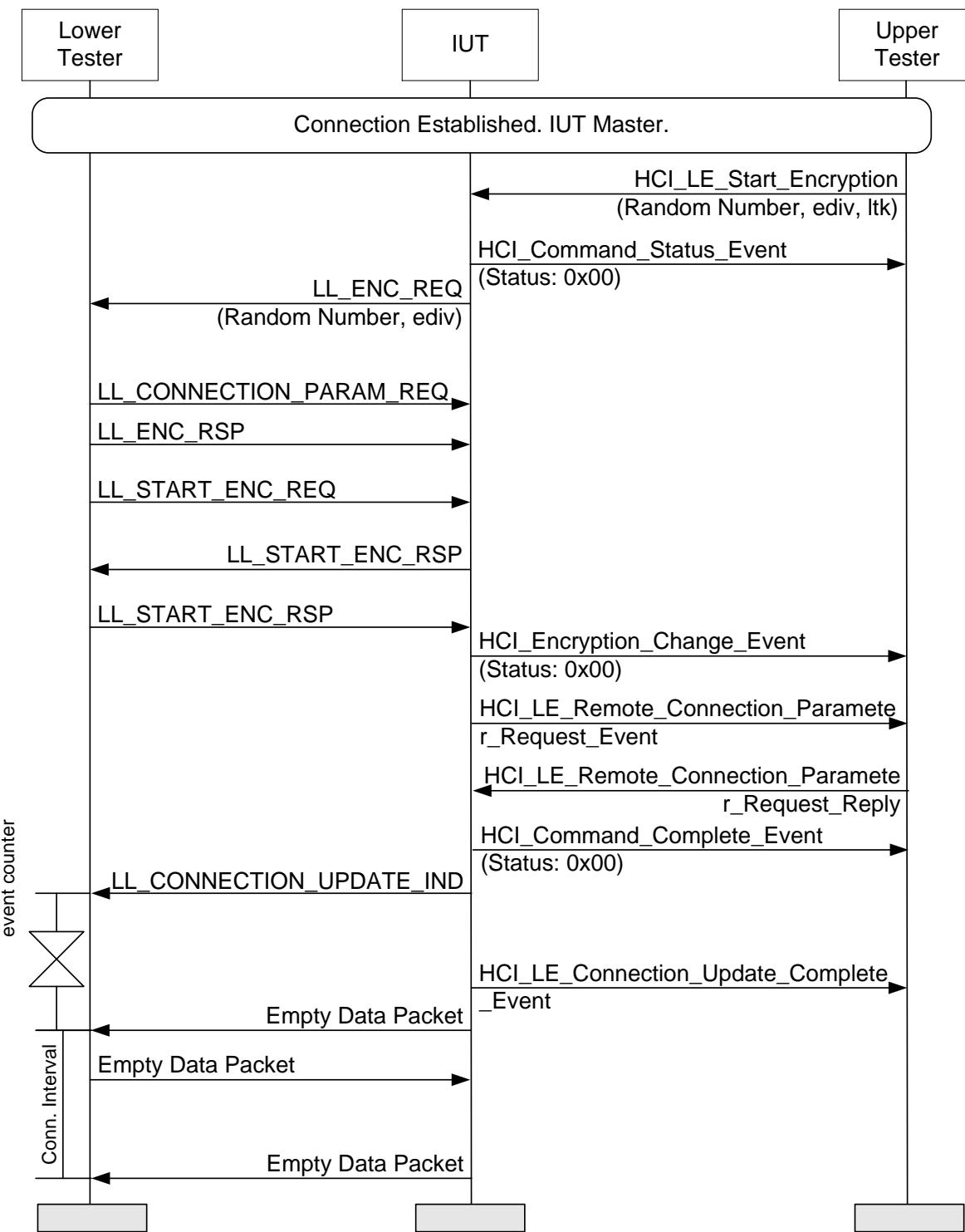


Figure 4.236: LL/CON/MAS/BV-28-C [Initiating Connection Parameter Request – different procedure collision – encryption]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT. Expect an HCI_Command_Status event from the IUT in response.
 2. Lower Tester expects an LL_ENC_REQ PDU from the IUT.

3. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT with connection parameters set to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) as an acknowledgement packet to the LL_ENC_REQ PDU.
 4. Lower Tester responds with an LL_ENC_RSP PDU.
 5. Lower Tester sends an LL_START_ENC_REQ to the IUT and expects the IUT to acknowledge it.
 6. Lower Tester expects an LL_START_ENC_RSP from the IUT and responds with an LL_START_ENC_RSP.
 7. The Upper Tester expects an HCI_Encryption_Change event from the IUT.
 8. At any time after step 3, and possibly interlaced with steps 4 to 7, the IUT requests the Upper Tester to accept or reject the Lower Tester's request to update the connection parameters. The Upper Tester accepts the request.
 9. Lower Tester expects the IUT to send an LL_CONNECTION_UPDATE_IND (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester).
 10. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
 11. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
 12. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT sends the LL_ENC_REQ PDU.

The IUT sends an LL_START_ENC_RSP PDU until acknowledged.

The IUT successfully reports the encryption change with the HCI event HCI_Encryption_Change.

The IUT responds positively to the Lower Tester's request to update connection parameters.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

- Notes

The Lower Tester and Upper Tester ensure that the encryption start procedure completes before the procedure response timeout for the connection parameter request procedure fires.

4.3.5.25 LL/CON/MAS/BV-29-C [Initiating Connection Parameter Request – remote legacy host]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device's host either does not support the connection parameter request procedure or has masked the remote connection parameters request event.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and rejects the IUT's request.

- Reference

[8] 5.1.7



- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).

- Test Procedure

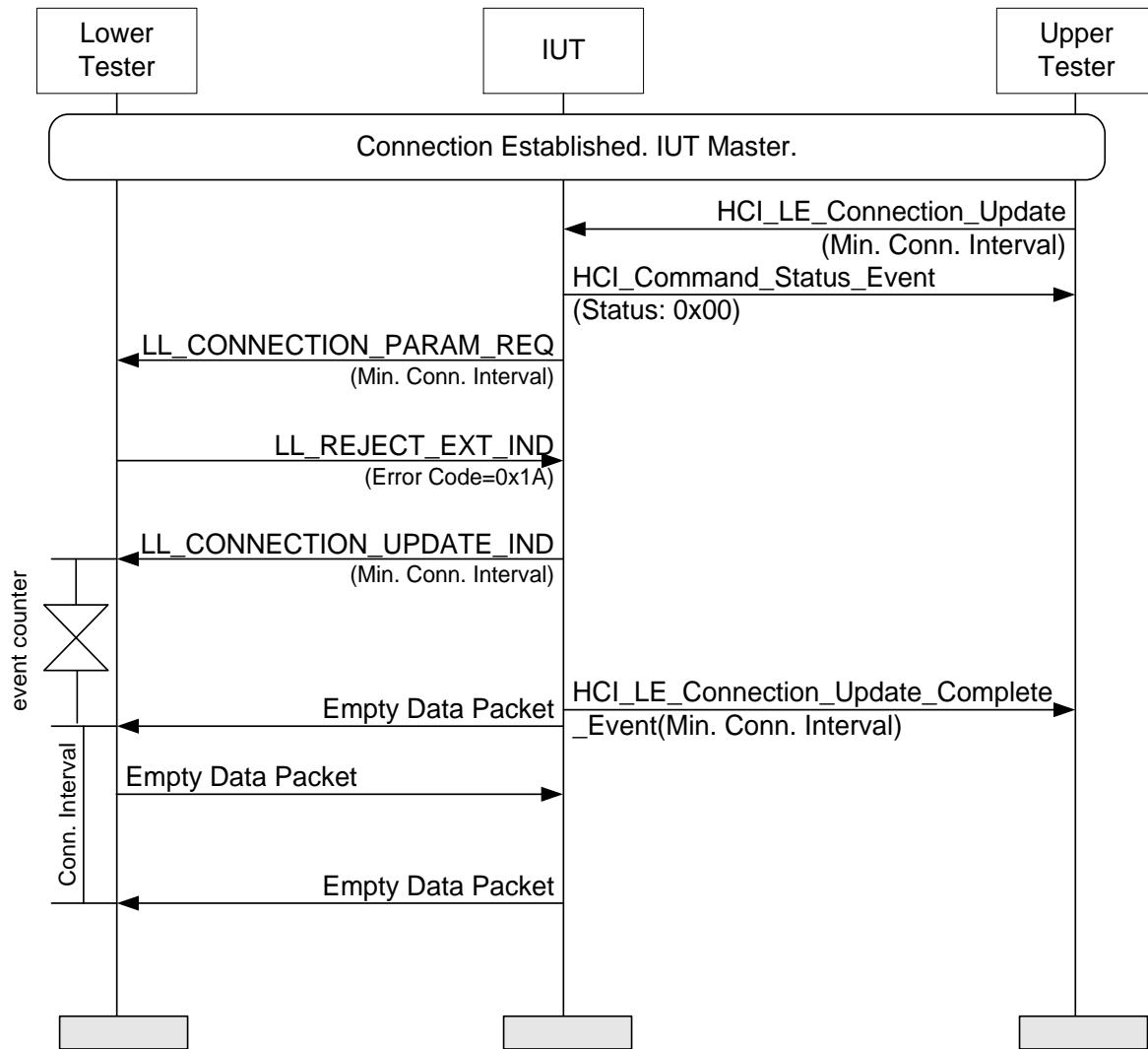


Figure 4.237: LL/CON/MAS/BV-29-C [Initiating Connection Parameter Request – remote legacy host]

- Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
- Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester rejects the IUT's request by issuing an

LL_REJECT_EXT_IND PDU containing ErrorCode 0x1A. Lower Tester expects a packet from the IUT acknowledging the LL_REJECT_EXT_IND PDU.

3. Lower Tester expects an LL_CONNECTION_UPDATE_IND control PDU from the IUT.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict

- The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters.
- The IUT sends an LL_CONNECTION_UPDATE_IND PDU to the Lower Tester when the Lower Tester sends an LL_REJECT_EXT_IND PDU with ErrorCode 0x1A in response to the IUT's LL_CONNECTION_PARAM_REQ PDU.
- The IUT maintains the connection with the new parameters selected by the IUT in use.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.26 LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred Periodicity]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave does not indicate any preferred periodicity.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

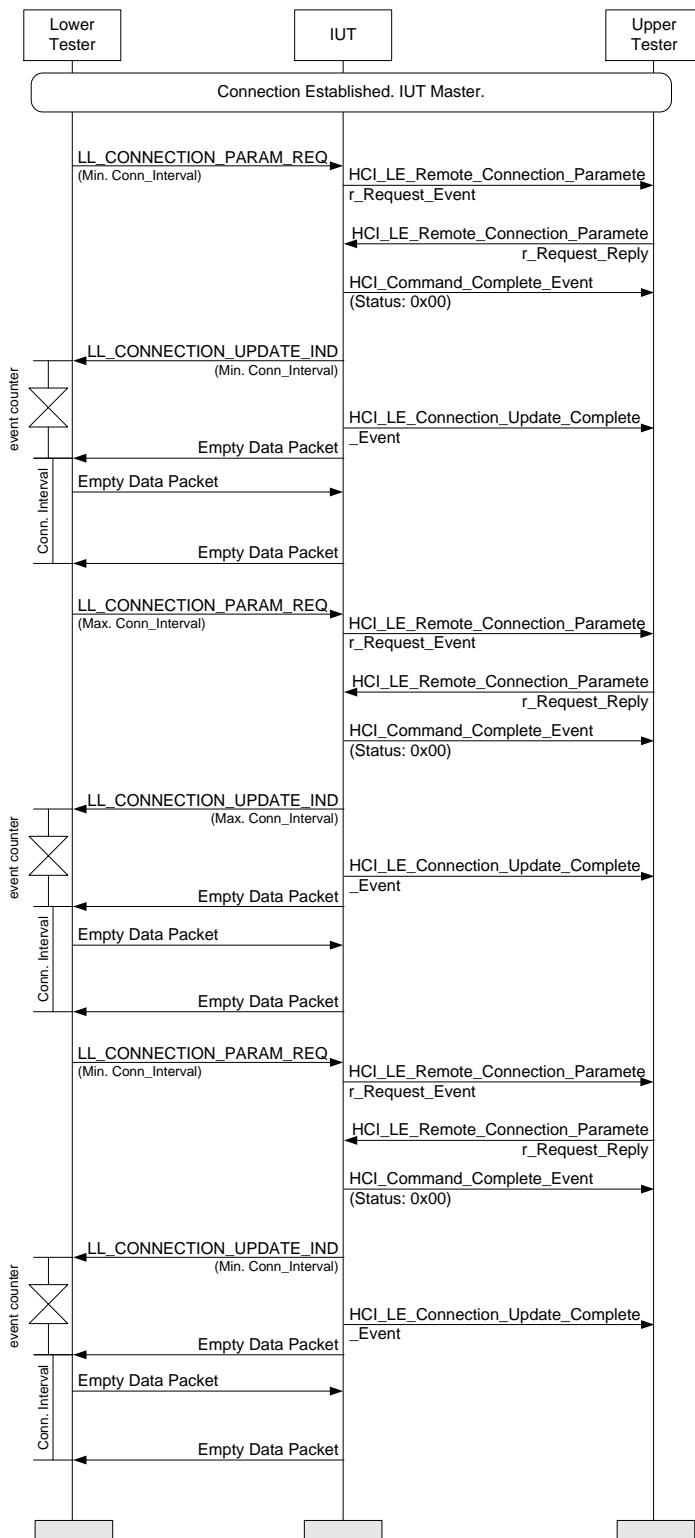


Figure 4.238: LL/CON/MAS/BV-30-C [Accepting Connection Parameter Request – no Preferred_Periodicity]



Case 1:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the maximum connection interval, no latency, and maximum connection supervision timeout and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and Preferred_Periodicity set to 0.
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. At the time of the update start maintaining the connection with the new parameters.
7. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.



- Expected Outcome

Pass Verdict

For all the three cases described in the test procedure, the following conditions shall occur:

- The IUT responds positively to the Lower Tester's request to update connection parameters.
- The IUT maintains the connection with the new parameters selected by the IUT in use.
- The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

4.3.5.27 LL/CON/MAS/BV-31-C [Accepting Connection Parameter Request – preferred anchor points only]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave only requests a change in anchor points.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

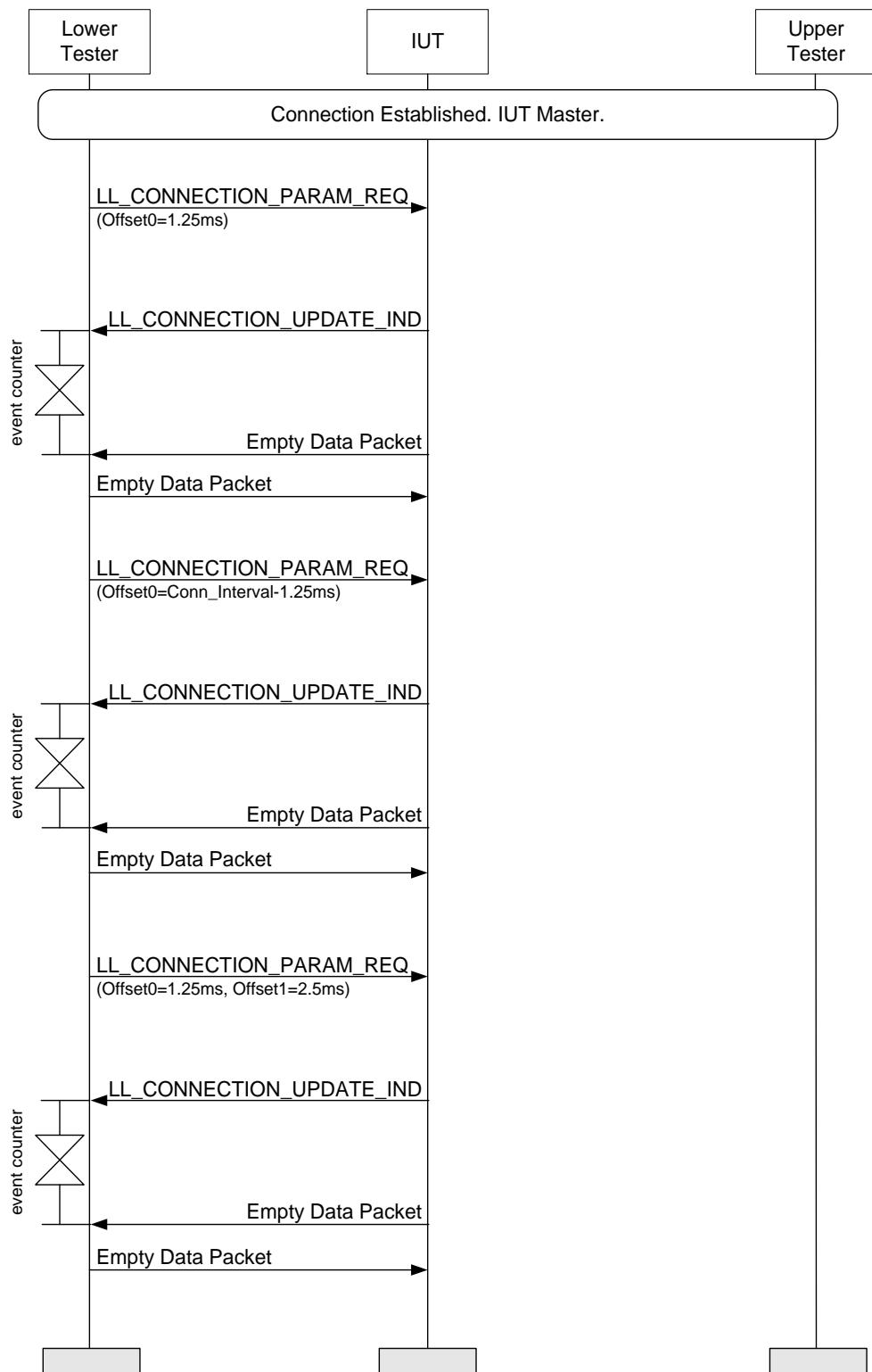


Figure 4.239: LL/CON/MAS/BV-31-C [Accepting Connection Parameter Request – preferred anchor points only]



Case 1:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are 1.25ms from the old anchor points. Lower Tester acknowledges the connection update request.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval will be used.

Case 2:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to (connection interval – 1.25ms), Offset1-5 to invalid and the connection interval, latency and supervision timeout unchanged.
2. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are (connection interval - 1.25ms) from the old anchor points. Lower Tester acknowledges the connection update request.
3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
4. Once the event count matches the time, the new parameters such as the new connection interval will be used.

Case 3:

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting Offset0 to 1.25ms, Offset1 to 2.5ms, Offset2-5 to invalid and the connection interval, latency and supervision timeout unchanged.
 2. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT such that the new anchor points are 1.25ms from the old anchor points. Lower Tester acknowledges the connection update request.
 3. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
 4. Once the event count matches the time, the new parameters such as the new connection interval will be used.
- Expected Outcome

Pass Verdict

- The IUT responds positively to the Lower Tester's request to update connection parameters.
- The IUT maintains the connection with the new parameters in use:
 - In the first case, the connection events are shifted by 1.25ms.
 - In the second case, the connection events are shifted by (connection interval – 1.25ms).
 - In the third case, the connection events are shifted by 1.25ms.



4.3.5.28 LL/CON/MAS/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave indicates a preferred periodicity.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

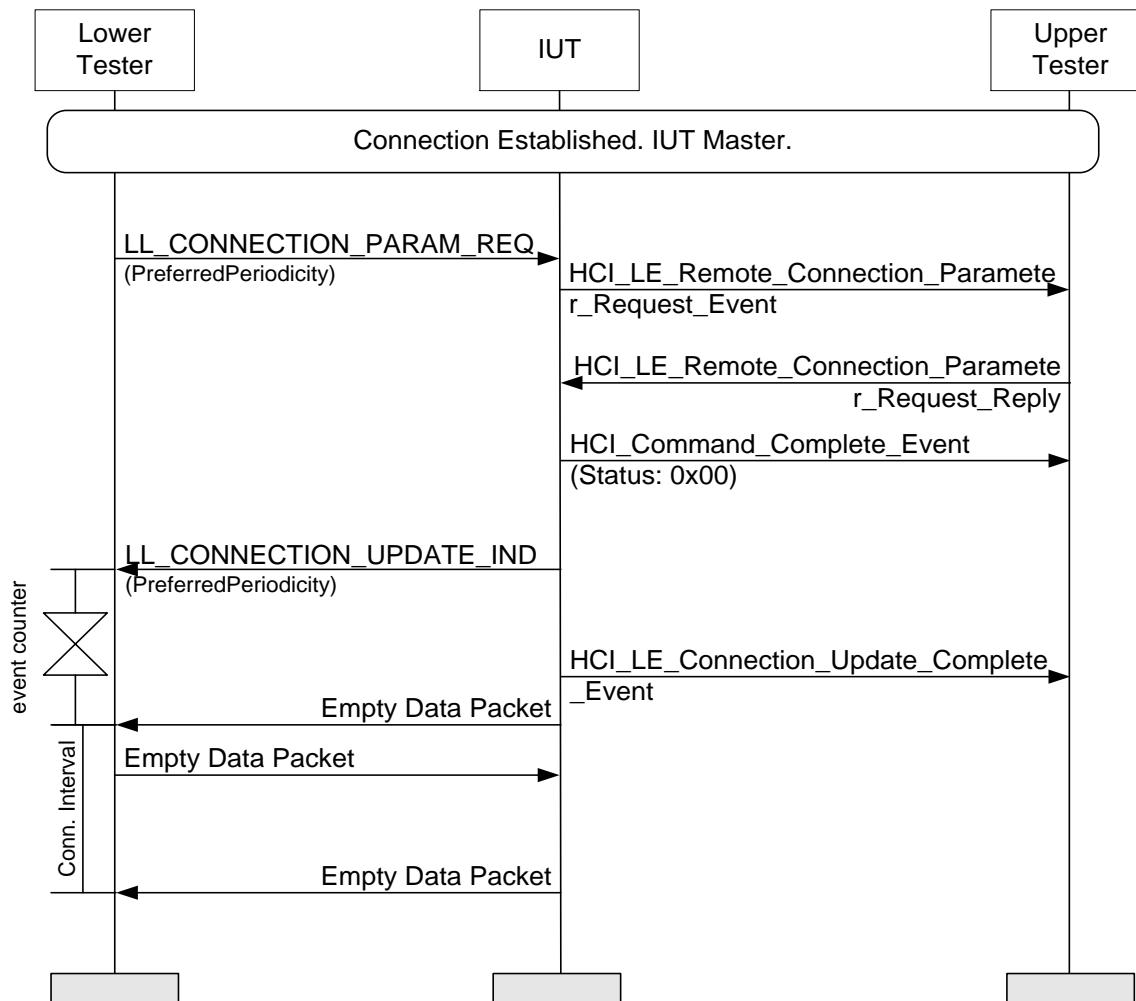


Figure 4.240: LL/CON/MAS/BV-32-C [Accepting Connection Parameter Request – Preferred_Periodicity]



1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s) and a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range.
 2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
 3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT containing a connection interval that is a multiple of the preferred periodicity indicated by the Lower Tester (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). Lower Tester acknowledges the connection update request.
 4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
 5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
 6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester's request to update connection parameters.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an
HCI_LE_Connection_Update_Complete event.

4.3.5.29 LL/CON/MAS/BV-33-C [Accepting Connection Parameter Request – Preferred_Periodicity and preferred anchor points]

- Test Purpose
Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave indicates a preferred periodicity and preferred anchor points.
- Reference
[\[8\] 5.1.7](#)
- Initial Conditions

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

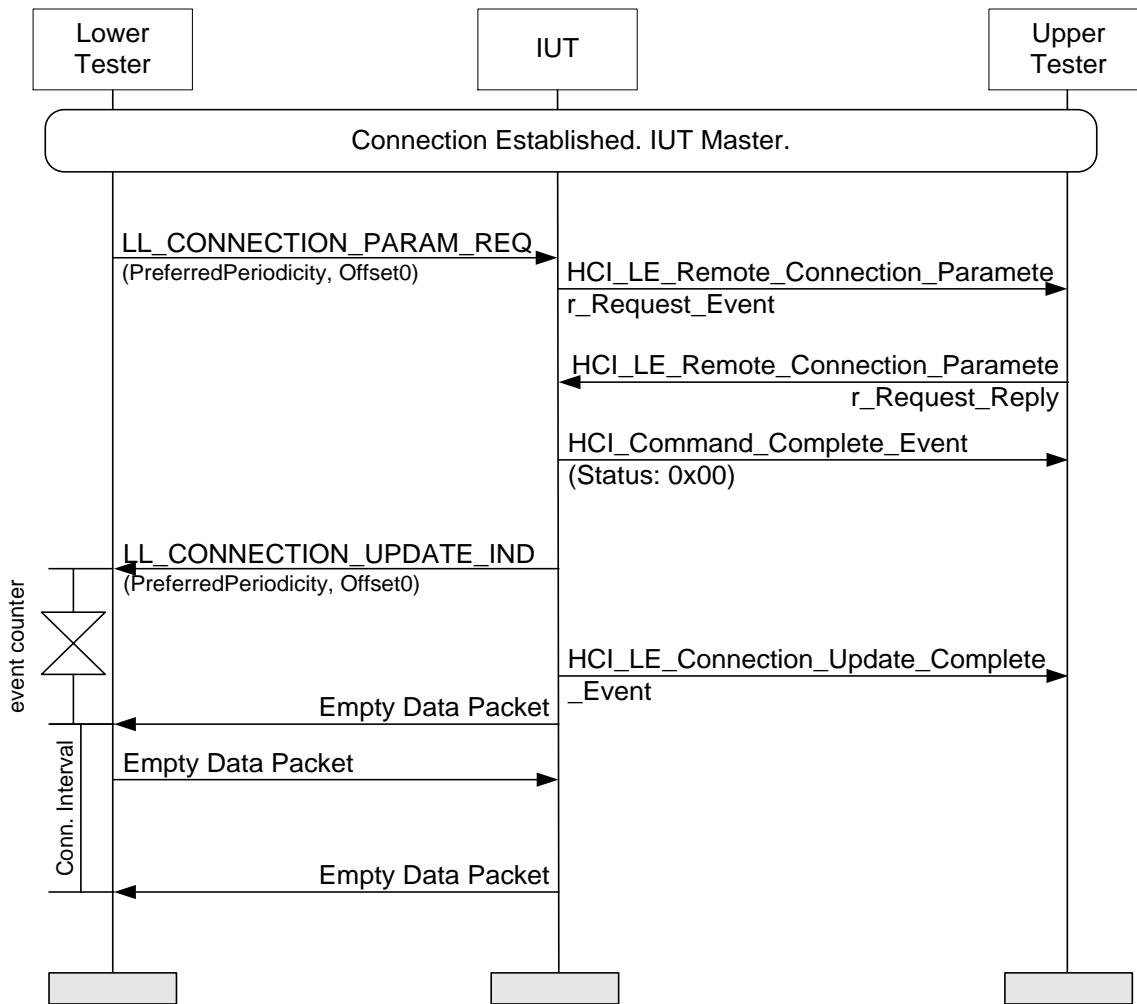


Figure 4.241: LL/CON/MAS/BV-33-C [Accepting Connection Parameter Request –Preferred_Periodicity and preferred anchor points]

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to a connection interval (with a non-zero range), no latency, intermediate connection supervision timeout (3 s), a preferred periodicity such that there is at least one connection interval that is a multiple of the preferred periodicity within the connection interval range, a reference connection event counter and a valid Offset0 value such that the new connection event is 1.25ms away from the old connection event at the reference connection event count (Offset1-5 are invalid).
2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester accepts the request.
3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT. Lower Tester acknowledges the connection update request.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict

The IUT responds positively to the Lower Tester's request to update connection parameters.

The IUT maintains the connection with the new parameters selected by the IUT in use.

The IUT reports the selected new connection parameters with an
HCI_LE_Connection_Update_Complete event.

4.3.5.30 LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave requires the master LL to request for approval from the master's Host and the master's Host has masked the LE Remote Connection Parameter Request Event.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).

- Test Procedure

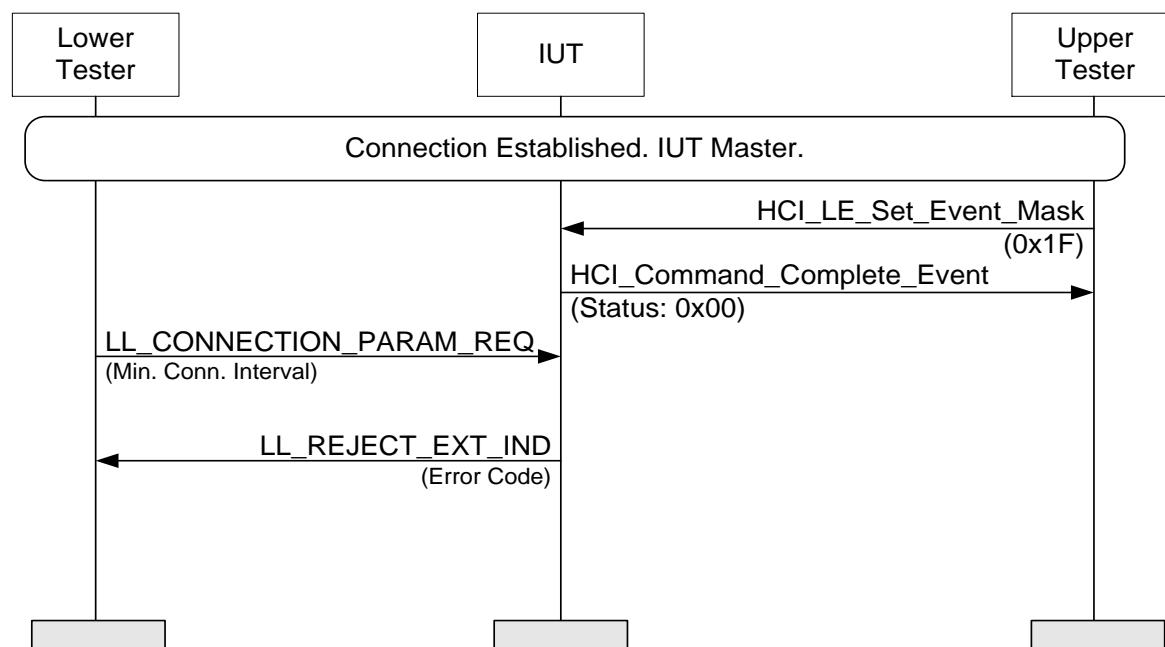


Figure 4.242: LL/CON/MAS/BV-34-C [Accepting Connection Parameter Request – event masked]



1. Upper Tester masks the LE Remote Connection Parameter event on the IUT.
 2. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
 3. Lower Tester expects an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1A.
- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode (0x1A).

4.3.5.31 LL/CON/MAS/BV-35-C [Accepting Connection Parameter Request – Host rejects]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the master's Host rejects the slave's connection parameter request procedure.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).

- Test Procedure

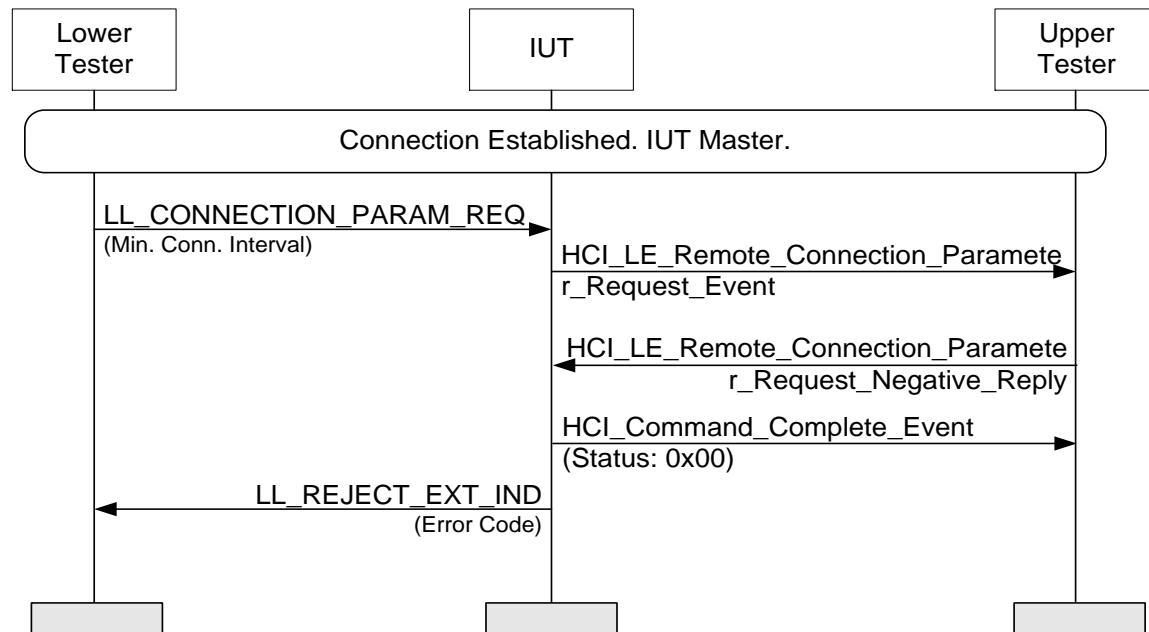


Figure 4.243: LL/CON/MAS/BV-35-C [Accepting Connection Parameter Request – Host rejects]



1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s).
 2. The IUT requests the Upper Tester to accept or reject the Lower Tester's request. The Upper Tester rejects the request using error code 0x3B.
 3. Lower Tester expects an LL_REJECT_EXT_IND control PDU from the IUT containing the ErrorCode provided by the Upper Tester.
- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode provided by the Upper Tester.

4.3.5.32 LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]

- Test Purpose

Test that a master IUT is able to perform the PHY update procedure. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT and accepts the IUT's request.

- Reference

[\[10\]](#) 5.1.10

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

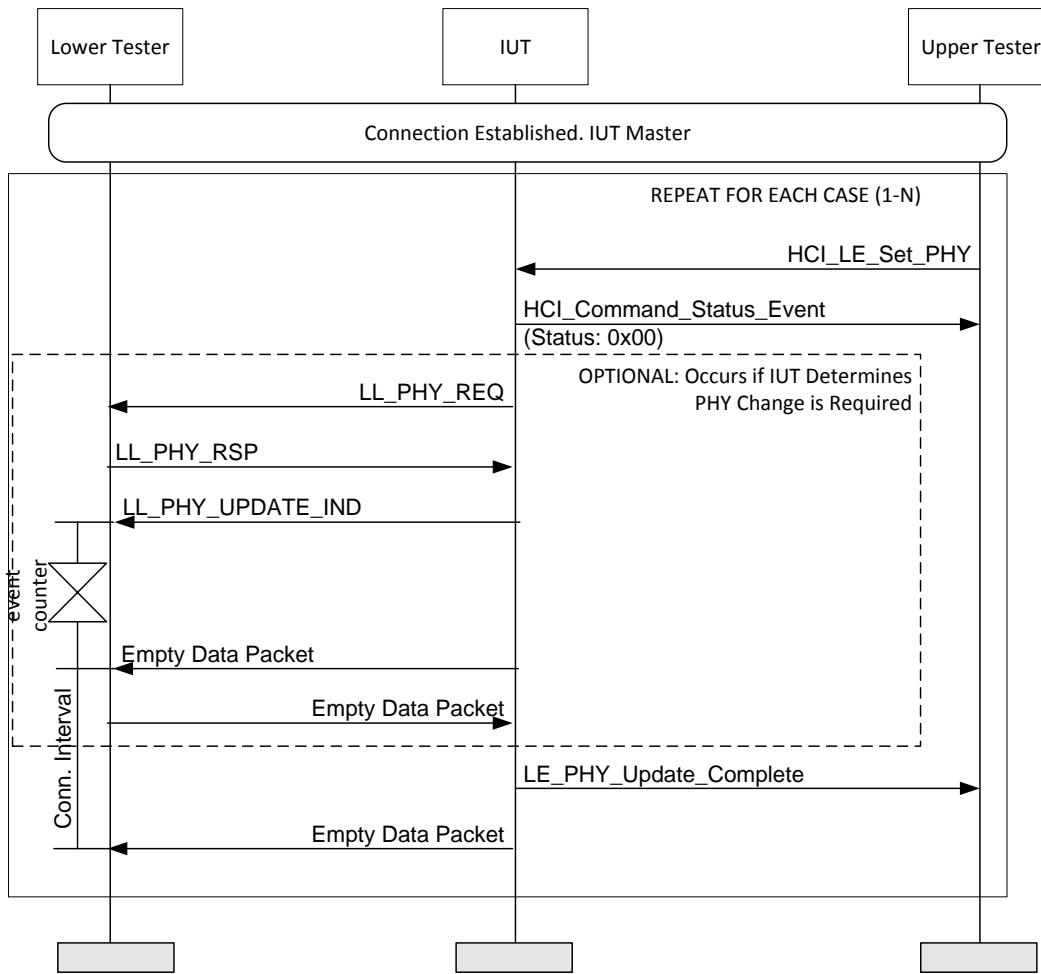


Figure 4.244: LL/CON/MAS/BV-41-C [Initiating PHY Update Procedure]

The following steps shall be carried out $2N$ times as follows, where N is the number of cases in [Table 4.37](#), [Table 4.38](#), or [Table 4.39](#) (selected based on the supported PHY(s)):

- firstly using cases 1 to N from the relevant table in order;
 - then using the cases from the relevant table in a random order.
1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with the payload defined in the **HCI_LE_Set_PHY** section of the relevant table and **PHY_options** set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
 2. If IUT initiates a PHY change, Lower Tester expects an **LL_PHY_REQ** control PDU from the IUT with at least one bit set in each field (RX_PHYS, TX_PHYS). Lower Tester acknowledges the IUT's request and responds with an **LL_PHY_RSP** PDU with the payload defined in the **LL_PHY_RSP** section of the relevant table.
 3. Lower Tester expects an **LL_PHY_UPDATE_IND** with zero or one bits set in each field(M_TO_S_PHY, S_TO_M_PHY) and a selected PHY present in the payload sent in the **LL_PHY_RSP** PDU. Proceed to step 4.
OR
 4. If IUT does NOT initiate a PHY change, proceed to step 7).
 5. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the **LL_PHY_UPDATE_IND** packet.

6. Once the event count matches the time, the new PHY(s) selected by the IUT will be used (or no change if both fields are zero).
7. IUT sends empty DATA packets to the Lower Tester, and Lower Tester acknowledges these packets, using the selected PHY(s).
8. Upper Tester expects a LE_PHY_Update_Complete event from the IUT with a payload consistent with the PHY(s) indicated in the LL_PHY_UPDATE_IND PDU (or the prior PHY, in cases where a field in LL_PHY_UPDATE_IND was zero or LL_PHY_UPDATE_IND was not sent).

Case	HCI_LE_Set_PHY			LL_PHY_RSP	
	ALL_PHYS	TX_PHYS	RX_PHYS	RX_PHYS	TX_PHYS
1	0x00	0x02	0x02	0x02	0x02
2	0x00	0x01	0x02	0x01	0x02
3	0x00	0x02	0x01	0x02	0x01
4	0x00	0x01	0x01	0x01	0x01
5	0x00	0x03	0x02	0x03	0x02
6	0x00	0x03	0x01	0x03	0x01
7	0x00	0x01	0x03	0x01	0x03
8	0x00	0x02	0x03	0x02	0x03
9	0x00	0x03	0x03	0x03	0x03
10	0x01	0x00	0x02	0x03	0x03
11	0x02	0x02	0x00	0x03	0x03
12	0x03	0x00	0x00	0x03	0x03

Table 4.37: PDU payload contents for each case variation for LE 2M PHY.

Case	HCI_LE_Set_PHY			LL_PHY_RSP	
	ALL_PHYS	TX_PHYS	RX_PHYS	RX_PHYS	TX_PHYS
1	0x00	0x04	0x04	0x04	0x04
2	0x00	0x01	0x04	0x01	0x04
3	0x00	0x04	0x01	0x04	0x01
4	0x00	0x01	0x01	0x01	0x01
5	0x00	0x05	0x04	0x05	0x04
6	0x00	0x05	0x01	0x05	0x01
7	0x00	0x01	0x05	0x01	0x05
8	0x00	0x04	0x05	0x04	0x05
9	0x00	0x05	0x05	0x05	0x05
10	0x01	0x00	0x04	0x05	0x05
11	0x02	0x04	0x00	0x05	0x05
12	0x03	0x00	0x00	0x05	0x05

Table 4.38: PDU payload contents for each case variation for LE Coded PHY.



Case	HCI_LE_Set_PHY			LL_PHY_RSP	
	ALL_PHYS	TX_PHYS	RX_PHYS	RX_PHYS	TX_PHYS
1	0x00	0x02	0x02	0x02	0x02
2	0x00	0x01	0x02	0x01	0x02
3	0x00	0x02	0x01	0x02	0x01
4	0x00	0x01	0x01	0x01	0x01
5	0x00	0x03	0x02	0x03	0x02
6	0x00	0x03	0x01	0x03	0x01
7	0x00	0x01	0x03	0x01	0x03
8	0x00	0x02	0x03	0x02	0x03
9	0x00	0x03	0x03	0x03	0x03
10	0x01	0x00	0x02	0x03	0x03
11	0x02	0x02	0x00	0x03	0x03
12	0x03	0x00	0x00	0x03	0x03
13	0x00	0x04	0x04	0x04	0x04
14	0x00	0x01	0x04	0x01	0x04
15	0x00	0x04	0x01	0x04	0x01
16	0x00	0x01	0x01	0x01	0x01
17	0x00	0x05	0x04	0x05	0x04
18	0x00	0x05	0x01	0x05	0x01
19	0x00	0x01	0x05	0x01	0x05
20	0x00	0x04	0x05	0x04	0x05
21	0x00	0x05	0x05	0x05	0x05
22	0x01	0x00	0x04	0x05	0x05
23	0x02	0x04	0x00	0x05	0x05
24	0x03	0x00	0x00	0x05	0x05
25	0x00	0x06	0x06	0x02	0x02
26	0x00	0x06	0x06	0x04	0x04
27	0x00	0x06	0x06	0x02	0x02
28	0x00	0x02	0x04	0x02	0x04
29	0x00	0x04	0x02	0x04	0x02
30	0x03	0x00	0x00	0x04	0x04
31	0x03	0x00	0x00	0x02	0x02
32	0x03	0x00	0x00	0x01	0x01

Table 4.39: PDU payload contents for each case variation for LE 2M and LE Coded PHYs.



- Expected Outcome

Pass Verdict

For all cases described in the test procedure, the following conditions shall occur:

- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in both the LL_PHY_REQ and LL_PHY_RSP PDUs, or zero in the following cases:
 - a. The LL_PHY_REQ and LL_PHY_RSP have no common PHY for that field.
 - b. The LL_PHY_REQ and LL_PHY_RSP both specify the current PHY (no change).
 - c. The LL_PHY_UPDATE_IND has a zero for both fields.
- If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.
- The IUT maintains the connection with the PHY(s) selected by the LL_PHY_UPDATE_IND, only changing PHY(s) if LL_PHY_UPDATE_IND PDU was sent and a change was indicated in the PDU.
- The IUT reports the selected PHY(s) with a LE_PHY_Update_Complete event, even if the PHY(s) did not change. The contents of the LE_PHY_Update_Complete event are consistent with the LL_PHY_UPDATE_IND if the PHY changed or the prior PHY if no change occurred.

Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.5.33 LL/CON/MAS/BV-42-C [Initiating PHY Update Procedure – Symmetric Only]

- Test Purpose

Test that a master IUT is able to perform the PHY update procedure when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT and accepts the IUT's request.

- Reference

[10] 5.1.10

- Initial Condition

Same as [LL/CON/MAS/BV-41-C \[Initiating PHY Update Procedure\]](#).

- Test Procedure

Same as [LL/CON/MAS/BV-41-C \[Initiating PHY Update Procedure\]](#) except that, in step 1, when ALL_PHYS is 0x00 and TX_PHYS does not equal RX_PHYS, the IUT shall return the error code Unsupported Feature or Parameter Value (0x11) to the Upper Tester (it may still initiate a PHY change).



- Expected Outcome

Pass Verdict

Same as [LL/CON/MAS/BV-41-C \[Initiating PHY Update Procedure\]](#). In addition:

- Each time the LL_PHY_REQ PDU is sent by the IUT, it shall have the same value in the TX_PHYS and RX_PHYS fields and this value shall have exactly one bit set.
- Each time the LL_PHY_UPDATE_IND PDU is sent by the IUT, it shall have the same value in the M_TO_S_PHY and S_TO_M_PHY fields.

Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.5.34 [LL/CON/MAS/BV-43-C \[Responding to PHY Update Procedure\]](#)

- Test Purpose

Test that a master IUT is able to respond to a PHY update procedure from a slave device. Test that the IUT can use all supported PHYs, including asymmetric settings. Test that the IUT successfully operates using the selected PHY(s).

- Reference

[\[10\]](#) 5.1.10

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

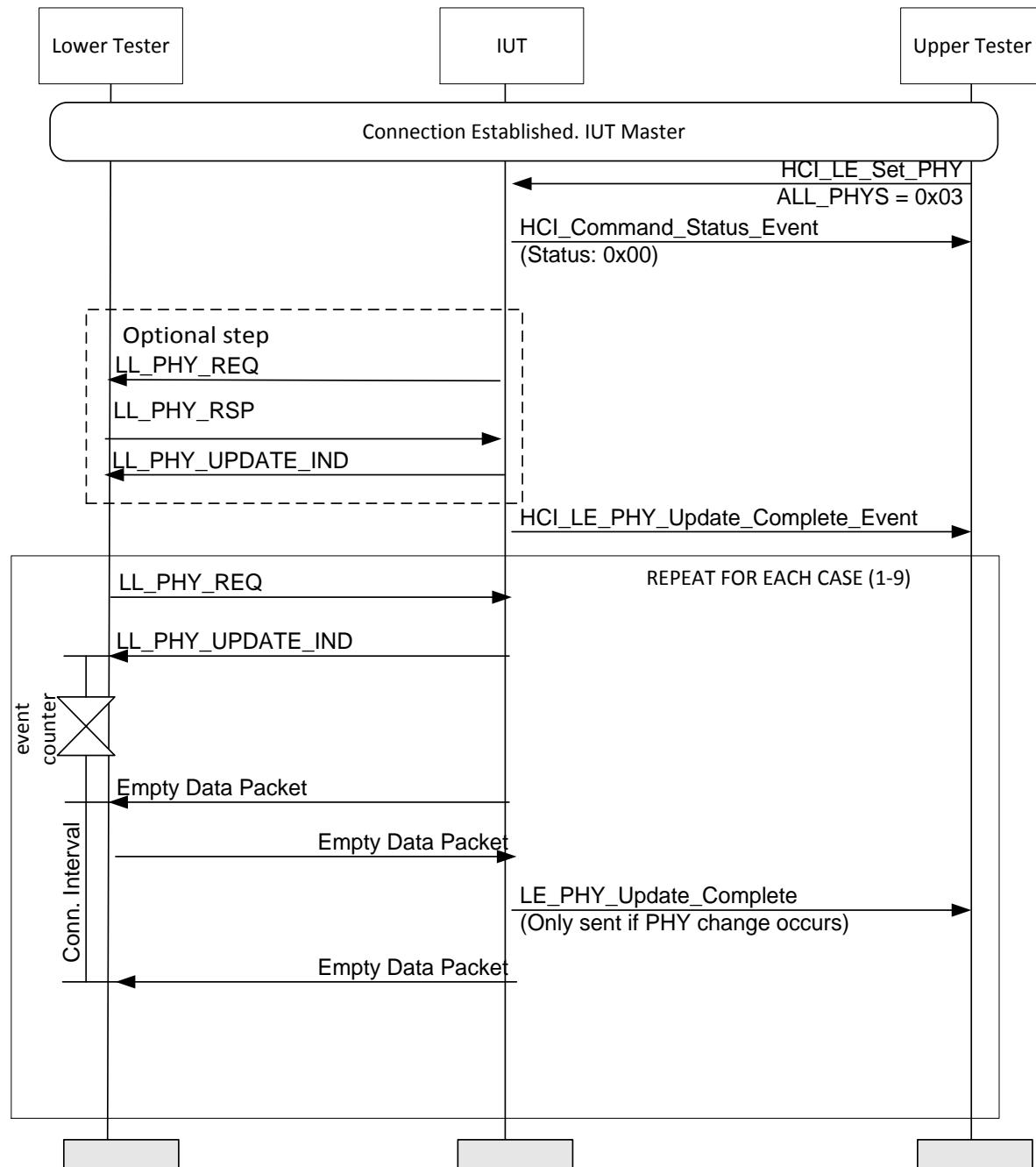


Figure 4.245: LL/CON/MAS/BV-43-C [Responding to PHY Update Procedure]

1. The Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with the **ALL_PHYS** field set to a value of 0x03. The Upper Tester expects an **HCI_Command_Status_Event** indicating success in response. The controller may send a **LL_PHY_REQ** to the Lower Tester. In this case, the Lower Tester sends a **LL_PHY_RSP** specifying the current PHY in both directions in response and the IUT completes the transaction with an **LL_PHY_UPDATE_IND**. Whether or not the procedure is carried out with the Lower Tester, the Upper Tester expects an **HCI LE PHY Update Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.

2. Perform steps 3 through 9 2N times as follows, where N is the number of cases in [Table 4.40](#), [Table 4.41](#), or [Table 4.42](#) (selected based on the supported PHY(s)):
 - firstly using cases 1 to N from the relevant table in order;
 - then using the cases from the relevant table in a random order.
3. Lower Tester sends an LL_PHY_REQ PDU to the IUT with the payload specified in the relevant table.
4. Lower Tester expects an LL_PHY_UPDATE_IND PDU from the IUT with a value selected for M_TO_S_PHY and S_TO_M_PHY that is either a bit value present in the LL_PHY_REQ or zero, with a maximum of 1 bit set for each field. If either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.
5. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
6. Once the event count matches the time, the PHY(s) selected by the IUT in the LL_PHY_UPDATE_IND packet will be used.
7. At the Instant of the PHY change start maintaining the connection with the selected PHY(s).
8. IUT sends empty DATA packets to the Lower Tester, and Lower Tester acknowledges these packets, using the selected PHY(s).
9. If the PHY(s) were changed, Upper Tester expects a LE_PHY_Update_Complete event from the IUT containing the PHY(s) selected. If both PHYs were NOT changed, Upper Tester expects NOT to receive a LE_PHY_Update_Complete event.

Case	LL_PHY_REQ	
	TX_PHYS	RX_PHYS
1	0x02	0x02
2	0x01	0x02
3	0x02	0x01
4	0x01	0x01
5	0x03	0x02
6	0x03	0x01
7	0x01	0x03
8	0x02	0x03
9	0x03	0x03

Table 4.40: PDU payload contents for each case variation for LE 2M PHY.

Case	LL_PHY_REQ	
	TX_PHYS	RX_PHYS
1	0x04	0x04
2	0x01	0x04
3	0x04	0x01
4	0x01	0x01
5	0x05	0x04
6	0x05	0x01
7	0x01	0x05
8	0x04	0x05



Case	LL_PHY_REQ	
	TX_PHYS	RX_PHYS
9	0x05	0x05

Table 4.41: PDU payload contents for each case variation for LE Coded PHY

Case	LL_PHY_REQ	
	TX_PHYS	RX_PHYS
1	0x03	0x01
2	0x05	0x02
3	0x02	0x04
4	0x01	0x02
5	0x04	0x01
6	0x03	0x06
7	0x01	0x01
8	0x04	0x03
9	0x05	0x01
10	0x04	0x04
11	0x05	0x07
12	0x05	0x05
13	0x04	0x02
14	0x03	0x07
15	0x06	0x06
16	0x03	0x02
17	0x01	0x06
18	0x05	0x06
19	0x04	0x05
20	0x01	0x05
21	0x05	0x03
22	0x01	0x04
23	0x01	0x03
24	0x03	0x05
25	0x06	0x04
26	0x02	0x07
27	0x06	0x01
28	0x02	0x02
29	0x03	0x04
30	0x07	0x03
31	0x02	0x01



Case	LL_PHY_REQ	
	TX_PHYS	RX_PHYS
32	0x03	0x03
33	0x02	0x03
34	0x04	0x07
35	0x07	0x04
36	0x07	0x01
37	0x06	0x05
38	0x02	0x06
39	0x07	0x07
40	0x04	0x06
41	0x02	0x05
42	0x06	0x02
43	0x07	0x02
44	0x07	0x06
45	0x06	0x07
46	0x06	0x03
47	0x05	0x04
48	0x07	0x05
49	0x01	0x07

Table 4.42: PDU payload contents for each case variation for LE 2M and LE Coded PHY.

- Expected Outcome

Pass Verdict

For all cases described in the test procedure, the following conditions shall occur:

- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in the LL_PHY_REQ, or zero. In addition, if the two fields of the LL_PHY_REQ held the same value which only had a single bit set, then the M_TO_S_PHY and S_TO_M_PHY fields shall either both contain that value or shall both be zero.
- If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.
- The IUT maintains the connection with the PHY(s) selected by the LL_PHY_UPDATE_IND payload.
- If the PHY(s) were changed, IUT sends a LE_PHY_Update_Complete event containing the PHY(s) selected in the LL_PHY_UPDATE_IND payload. If both PHYs were NOT changed, IUT does NOT send a LE_PHY_Update_Complete event.

Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case.



4.3.5.35 LL/CON/MAS/BV-44-C [Responding to PHY Update Procedure – Symmetric Only]

- Test Purpose

Test that a master IUT is able to respond to a PHY update procedure from a slave device when asymmetric links are not supported. Test that the IUT only requests symmetric PHY settings at a single rate. Test that the IUT successfully operates using the selected PHY(s).

- Reference

[10] 5.1.10

- Initial Condition

Same as [LL/CON/MAS/BV-43-C \[Responding to PHY Update Procedure\]](#).

- Test Procedure

Same as [LL/CON/MAS/BV-43-C \[Responding to PHY Update Procedure\]](#).

- Expected Outcome

Pass Verdict

Same as [LL/CON/MAS/BV-43-C \[Responding to PHY Update Procedure\]](#). In addition:

- Each time the LL_PHY_UPDATE_IND PDU is sent by the IUT, it shall have the same value in the M_TO_S_PHY and S_TO_M_PHY fields.

Inconclusive Verdict

The IUT does not initiate at least one PHY Update Procedure during this test case.

4.3.5.36 LL/CON/MAS/BV-45-C [Handling Protocol Collision – Same Procedure]

- Test Purpose

Test that a master IUT is able to perform the PHY update procedure when there is a procedure collision between the IUT's PHY change request and the remote device's PHY change request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY change request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT's PHY change request to cause a procedure collision. The test case expects the IUT to reject the slave's PHY change request and then proceed with its own PHY change request procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

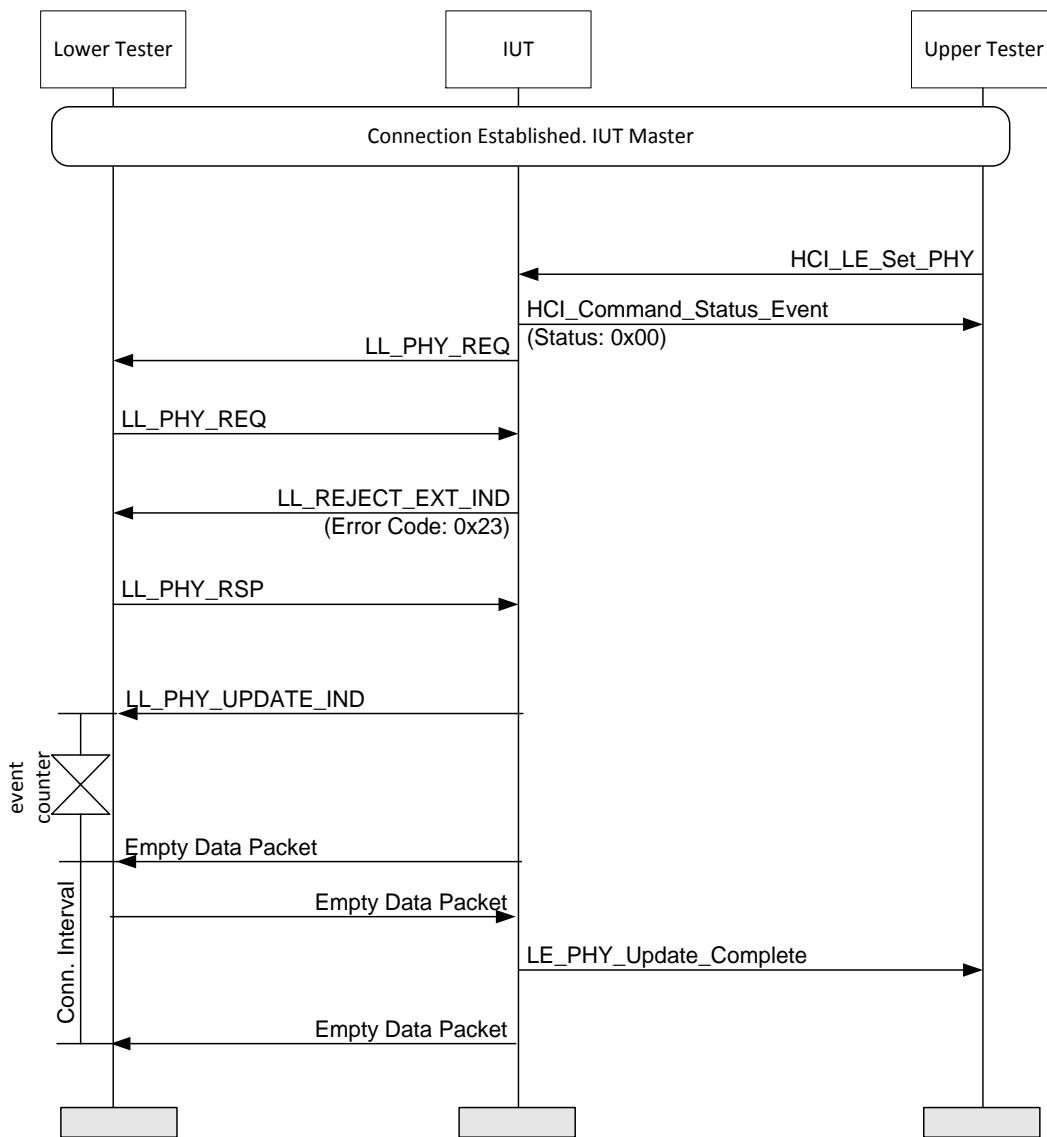


Figure 4.246: LL/CON/MAS/BV-45-C [Handling Protocol Collision – Same Procedure]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS set to prefer a PHY other than LE 1M and PHY_options set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
- Lower Tester expects an **LL_PHY_REQ** control PDU from the IUT with at least one bit set in each field (RX_PHYS, TX_PHYS). If the IUT does not send an **LL_PHY_REQ** PDU the test case ends with an Inconclusive Verdict.
- Lower Tester responds with an **LL_PHY_REQ** with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M.
- Lower Tester expects the IUT to reject the Lower Tester's **LL_PHY_REQ** using an **LL_REJECT_EXT_IND** with reason code 0x23.
- Lower Tester responds to the IUT's **LL_PHY_REQ** using an **LL_PHY_RSP** PDU with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M and expects the IUT to respond with an **LL_PHY_UPDATE_IND** with a value selected for M_TO_S_PHY and S_TO_M_PHY that is either a bit value present in the **LL_PHY_REQ** or zero, with a maximum of 1 bit set for each

- field. If either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value. If both fields are zero the test case ends with an Inconclusive Verdict.
6. Maintain the connection using empty DATA packets until the event count matches the Instant indicated in the LL_PHY_UPDATE_IND packet.
 7. Once the event count matches the time, the PHY(s) selected by the IUT in the LL_PHY_UPDATE_IND will be used (or prior PHY(s) if no change specified).
 8. At the Instant of the PHY change start maintaining the connection with the selected PHY(s).
 9. IUT sends empty DATA packets to the Lower Tester, and Lower Tester acknowledges these packets, using the selected PHY(s).
 10. Upper Tester expects a LE_PHY_Update_Complete from the IUT with values corresponding to the PHY(s) selected in the LL_PHY_UPDATE_IND PDU, or the prior PHY(s) if no change specified.
- Expected Outcome
- Pass Verdict
- If the IUT transmits an LL_PHY_UPDATE_IND PDU to update the selected PHY, the value chosen for M_TO_S_PHY and S_TO_M_PHY shall have a maximum of 1 bit set for each field. The PHY selected for each field must either be a PHY specified in the LL_PHY_RSP, or zero in the following cases:
 - a. The LL_PHY_REQ and LL_PHY_RSP have no common PHY for that field.
 - b. The LL_PHY_REQ and LL_PHY_RSP both specify the current PHY (no change).
 - c. The LL_PHY_RSP had the same single PHY selected for both directions and the LL_PHY_UPDATE_IND has a zero for both fields.
 - If the IUT transmits an LL_PHY_UPDATE_IND PDU where either the M_TO_S_PHY or S_TO_M_PHY fields are non-zero, then the Instant shall have a valid value.
 - The IUT reports the selected PHY(s) with a LE_PHY_Update_Complete event to the host, even if no change occurs.

Inconclusive Verdict

The IUT does not initiate the PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2), or does not select a PHY change in step 5)

4.3.5.37 LL/CON/MAS/BV-46-C [Protocol Timeout for PHY Update Procedure]

- Test Purpose

Test that a master IUT terminates the Link Layer connection if the master-initiated PHY update procedure is not completed before the procedure response timer expires.

The Lower Tester acts in the slave role in the connection and ensures that the procedure initiated by the IUT is not completed.

- Reference

[10] 5.2

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

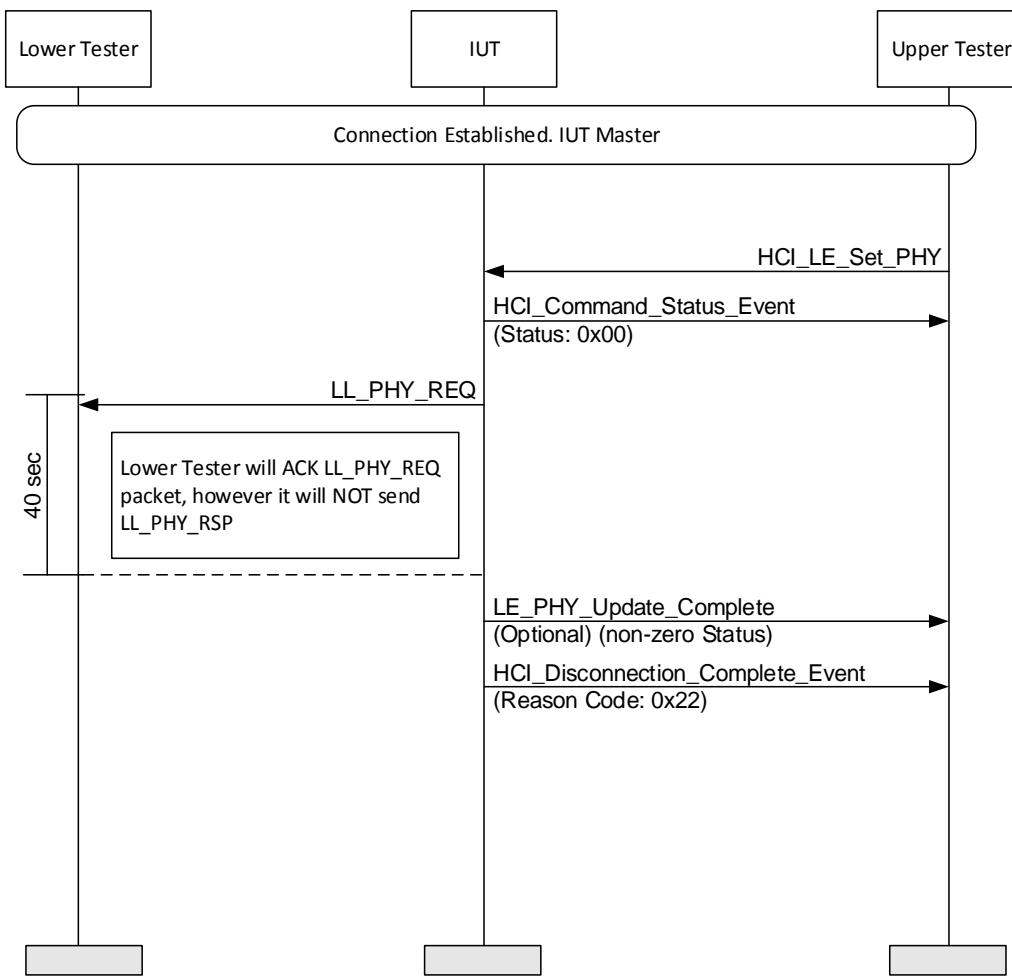


Figure 4.247: LL/CON/MAS/BV-46-C [Protocol Timeout for PHY Update Procedure]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with both fields (TX_PHYS, RX_PHYS) set to prefer a PHY other than LE 1M and PHY_options set to 0x0000 and receives an **HCI_Command_Status** event.
 - The IUT sends an **LL_PHY_REQ** PDU with at least one bit set of each field (RX_PHYS, TX_PHYS). If the IUT does not send an **LL_PHY_REQ** PDU the test case ends with an Inconclusive Verdict.
 - Lower Tester acknowledges the **LL_PHY_REQ** PDU but does not send an **LL_PHY_RSP** PDU.
 - The Upper Tester optionally expects the IUT to send an **HCI_LE_PHY_Update_Complete** event with a non-zero status.
 - IUT sends the **HCI_Disconnection_Complete** event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.
- Expected Outcome

Pass Verdict

The IUT sends **HCI_Disconnection_Complete** event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

If the IUT sends an **HCI_LE_PHY_Update_Complete** event to the Upper Tester in Step 4 the status shall be non-zero.



Inconclusive Verdict

The IUT does not initiate the PHY Update Procedure by sending an LL_PHY_REQ PDU in step 2).

4.3.5.38 LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]

- Test Purpose

Test that a master IUT is able to perform the channel map update procedure when there is a procedure collision between the IUT's channel map update and the Lower Tester's PHY change request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the channel map update procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT's channel map update to cause a procedure collision. The test case expects the IUT to reject the slave's PHY change request and then proceed with its own channel map update procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

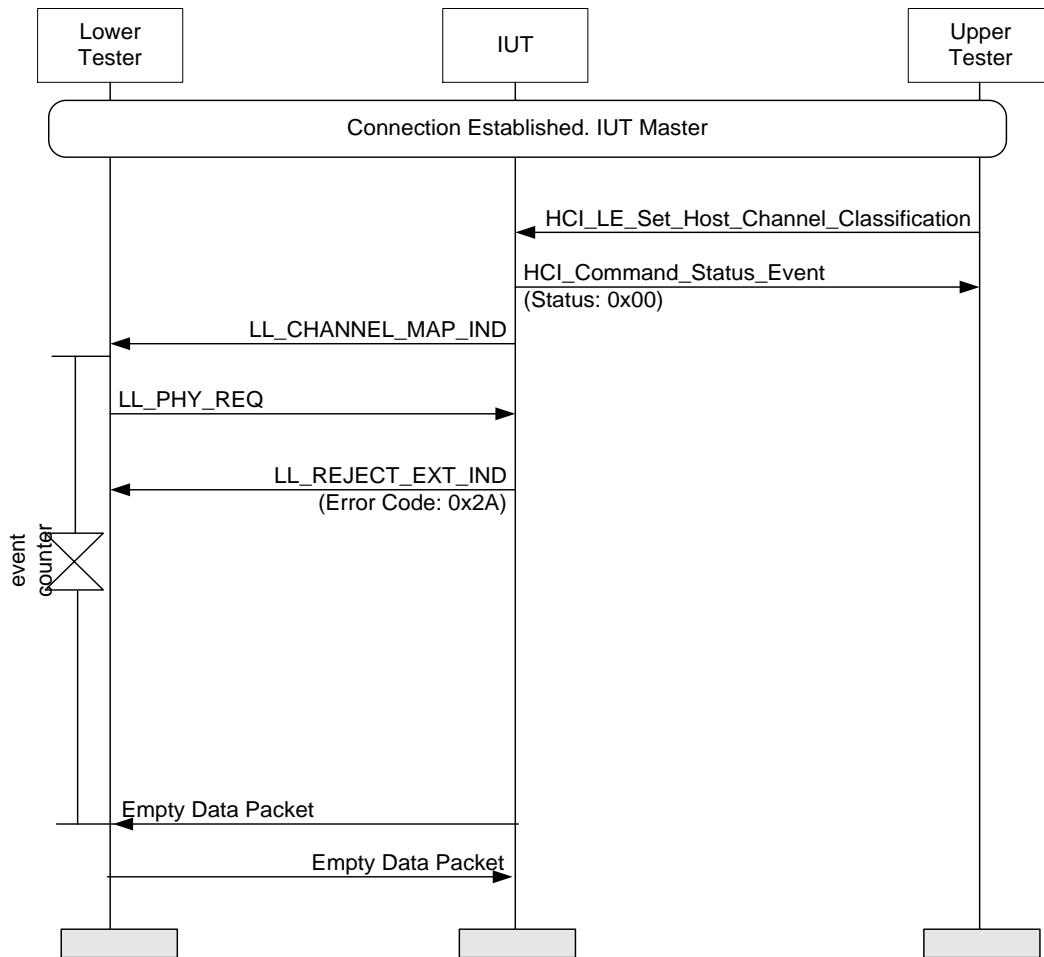


Figure 4.248: LL/CON/MAS/BV-47-C [Handling Protocol Collision – Different Procedure – Channel Map]

- Upper Tester sends an `HCI_LE_Set_Host_Channel_Classification` command to the IUT setting the channel map to only use even channels. Upper Tester expects an `HCI_Command_Complete` event from the IUT in response.
 - Lower Tester expects `LL_CHANNEL_MAP_IND` control PDU from the IUT, with the parameters submitted in step 1.
 - Lower Tester responds with an `LL_PHY_REQ` with both fields (`RX_PHYS`, `TX_PHYS`) set to prefer a PHY other than LE 1M.
 - Lower Tester expects the IUT to reject the Lower Tester's `LL_PHY_REQ` using an `LL_REJECT_EXT_IND` with reason code `0x2A`.
 - Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the channel map update request.
- Expected Outcome

Pass Verdict

The IUT transmits the `LL_CHANNEL_MAP_IND` PDU to update the channel map and rejects the Lower Tester's request to update the selected PHY(s).

The IUT maintains the connection with the Lower Tester with the updated data channel selection parameters after the assigned event.



4.3.5.39 LL/CON/MAS/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

- Test Purpose

Test that a master IUT is able to perform the connection parameters request procedure when there is a procedure collision between the IUT's connection parameters request and the Lower Tester's PHY change request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameters request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and initiates a new PHY change request procedure upon receiving the IUT's connection parameters request to cause a procedure collision. The test case expects the IUT to reject the slave's PHY change request and then proceed with its own connection parameters request procedure.

- Reference

[10] 5.3

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

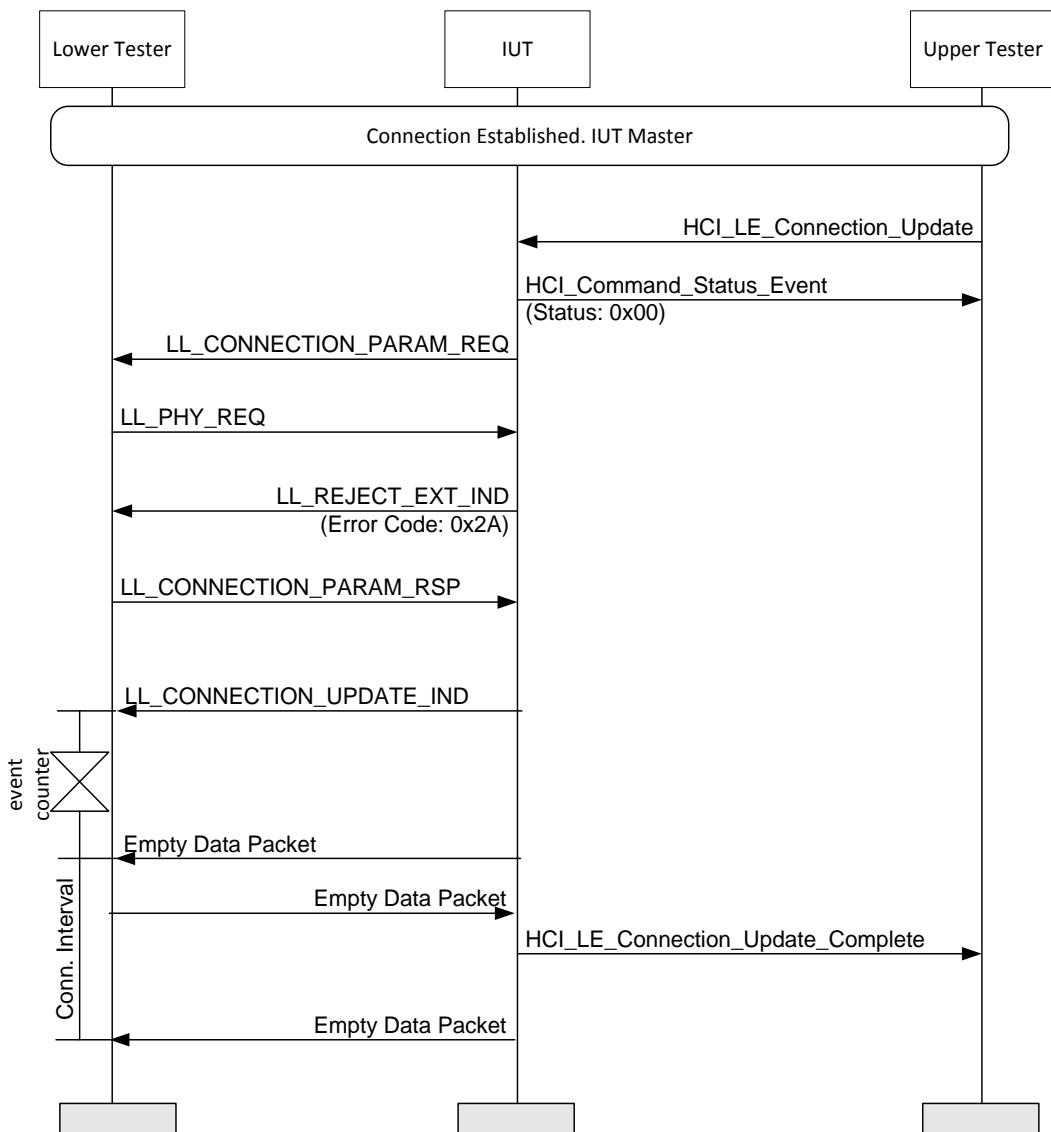


Figure 4.249: LL/CON/MAS/BV-48-C [Handling Protocol Collision – Different Procedure – Connection Parameters]

- Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
- Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL_PHY_REQ with both fields (RX_PHYS, TX_PHYS) set to prefer a PHY other than LE 1M.
- Lower Tester expects the IUT to reject the Lower Tester's LL_PHY_REQ using an LL_REJECT_EXT_IND with reason code 0x2A.
- Lower Tester responds to the IUT's connection parameters request using an LL_CONNECTION_PARAM_RSP and expects the IUT to respond with an LL_CONNECTION_UPDATE_IND.
- Maintain the connection using empty DATA packets. Repeat until the event count matches the time indicated in the connection update request packet.

6. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
 7. At the time of the update start maintaining the connection with the new parameters.
 8. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.
- Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters and rejects the Lower Tester's request to update the selected PHY(s).

The IUT maintains the connection with the new parameters selected by the IUT in use

The IUT reports the selected new connection parameters with an
HCI_LE_Connection_Update_Complete event.

4.3.5.40 LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions]

- Test Purpose

Test that a master IUT follows all packet time restrictions both during and after PHY change when it initiates the PHY update procedure.

The Lower Tester acts in the slave role maintaining a connection. A PHY update procedure is performed to set both direction to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queuing data to the IUT and issues the HCI command to start the PHY update procedure as the Host of the IUT. The Lower Tester observes the PHY update procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the procedure.

- Reference

[\[10\]](#) 5.1.10.1

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

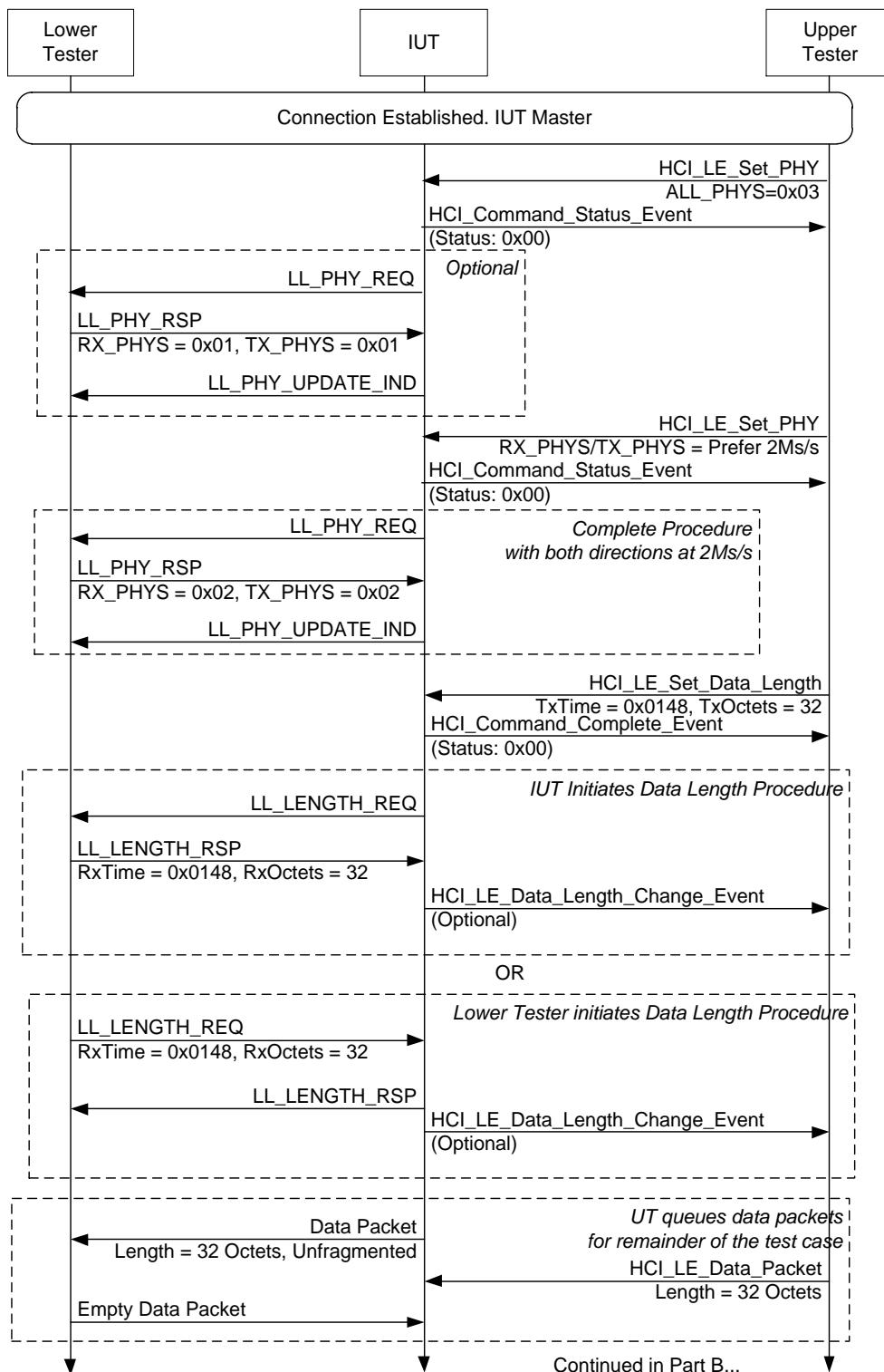


Figure 4.250: LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions – Part A]



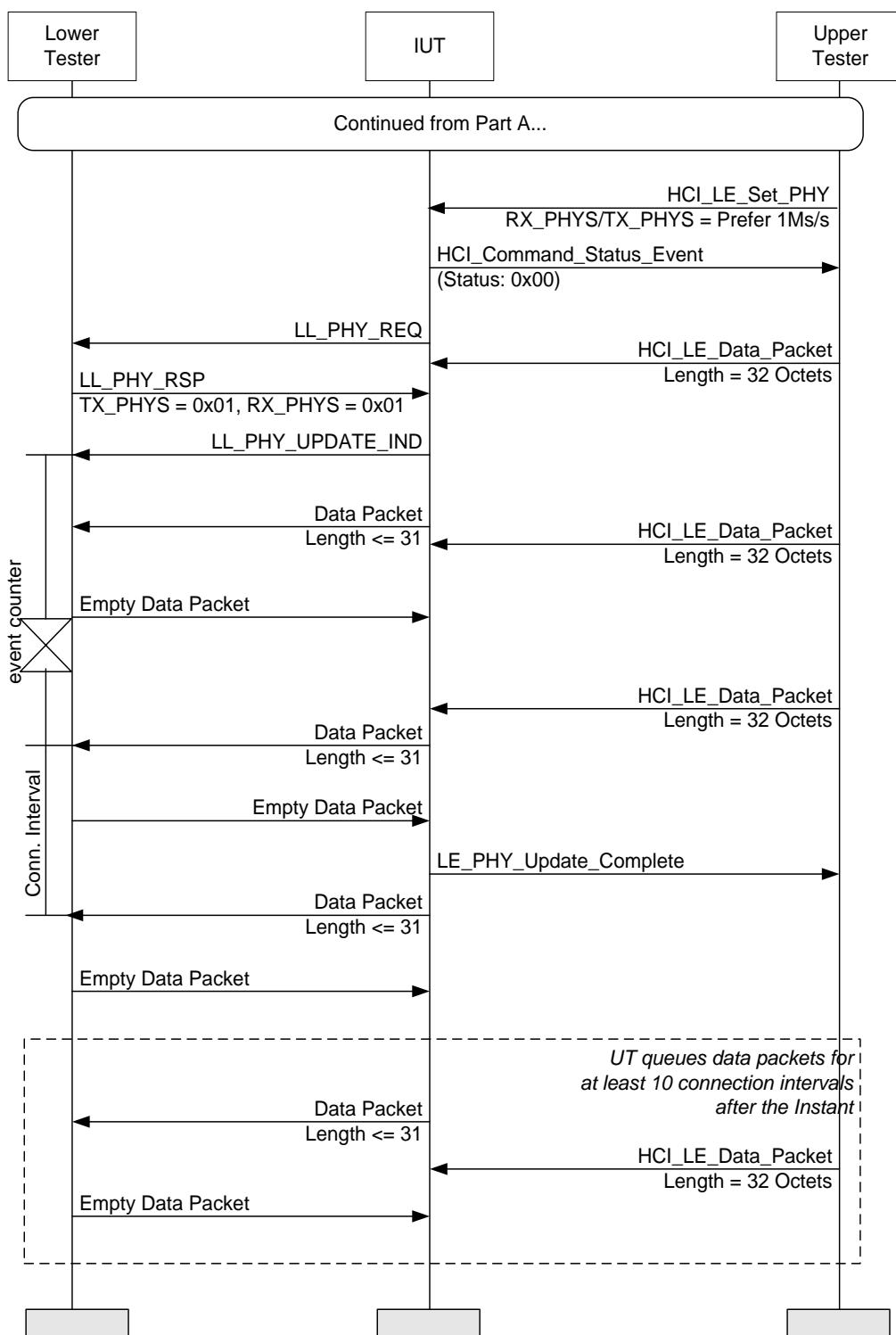


Figure 4.251: LL/CON/MAS/BV-49-C [Initiating PHY Update Procedure – Packet Time Restrictions - Part B]

1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with ALL_PHYS set to 0x03, and expects an **HCI_Command_Status** event from the IUT in response.
2. The Upper Tester expects an **LE_PHY_Update_Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.

3. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 2M PHY and expects an HCI_Command_Status event from the IUT in response.
 4. The Lower Tester expects an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_PSP with the RX_PHYS and TX_PHYS fields both set to 0x02.
 5. Lower Tester expects an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE 2M PHY for both directions. If the IUT does not select the LE 2M PHY for both directions the test case ends with an Inconclusive Verdict.
 6. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
 7. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and expects an HCI_Command_Complete event from the IUT in response.
 8. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
 9. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 10. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 11. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 1M PHY.
 12. The Lower Tester expects an LL_PHY_REQ from the IUT and responds with an LL_PHY_RSP with the RX_PHYS and TX_PHYS fields both set to 0x01.
 13. The Lower Tester expects an LL_PHY_UPDATE_IND from the IUT selecting the LE 1M PHY to be used in both directions. If the IUT does not select the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.
 14. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 13) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY update procedure.
 15. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 16. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.
 17. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
 18. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in Step 13) must have a length of 32 octets and be un-fragmented.



All data packets received by the Lower Tester after the IUT sends an LL_PHY_LL_PHY_UPDATE_IND in Step 13) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 1M PHY for both directions in Step 2).
- The IUT does not select the LE 2M PHY for both directions in Step 5).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 8).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in Step 13).
- The IUT does not select the LE 1M PHY for both directions in Step 13).

4.3.5.41 LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions]

- **Test Purpose**

Test that a master IUT both during and after PHY change follows all packet time restrictions when it responds to a PHY update procedure from a slave device.

The Lower Tester acts in the slave role maintaining a connection. A PHY update procedure is performed to set both directions to the LE 2M PHY and a data length update procedure is performed. The Upper Tester begins queueing data to the IUT and starts the PHY update procedure. The Lower Tester observes the PHY update procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the change.

- **Reference**
- [10] 5.1.10.1
- **Initial Condition**

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

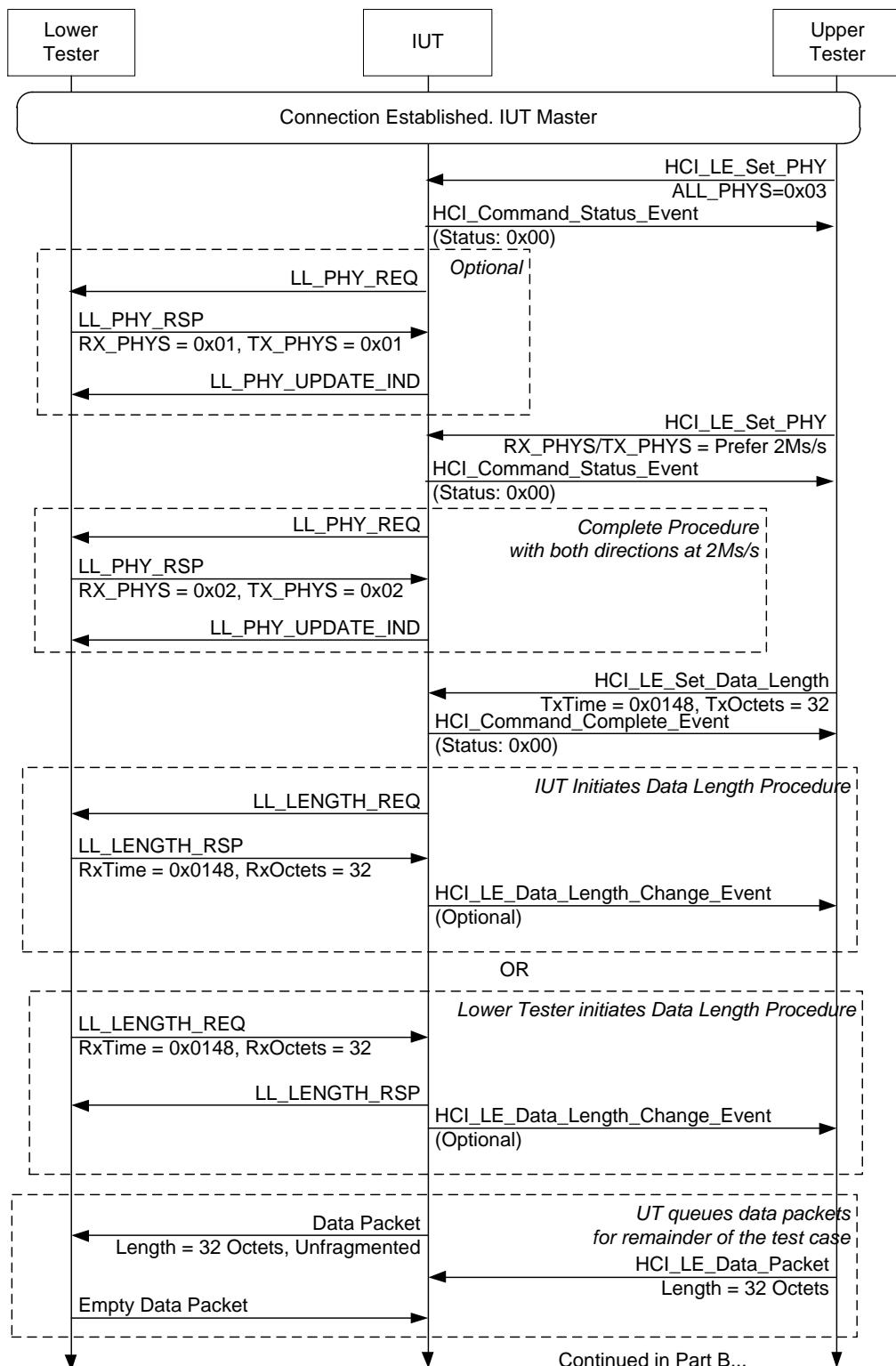


Figure 4.252: LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions – Part A]



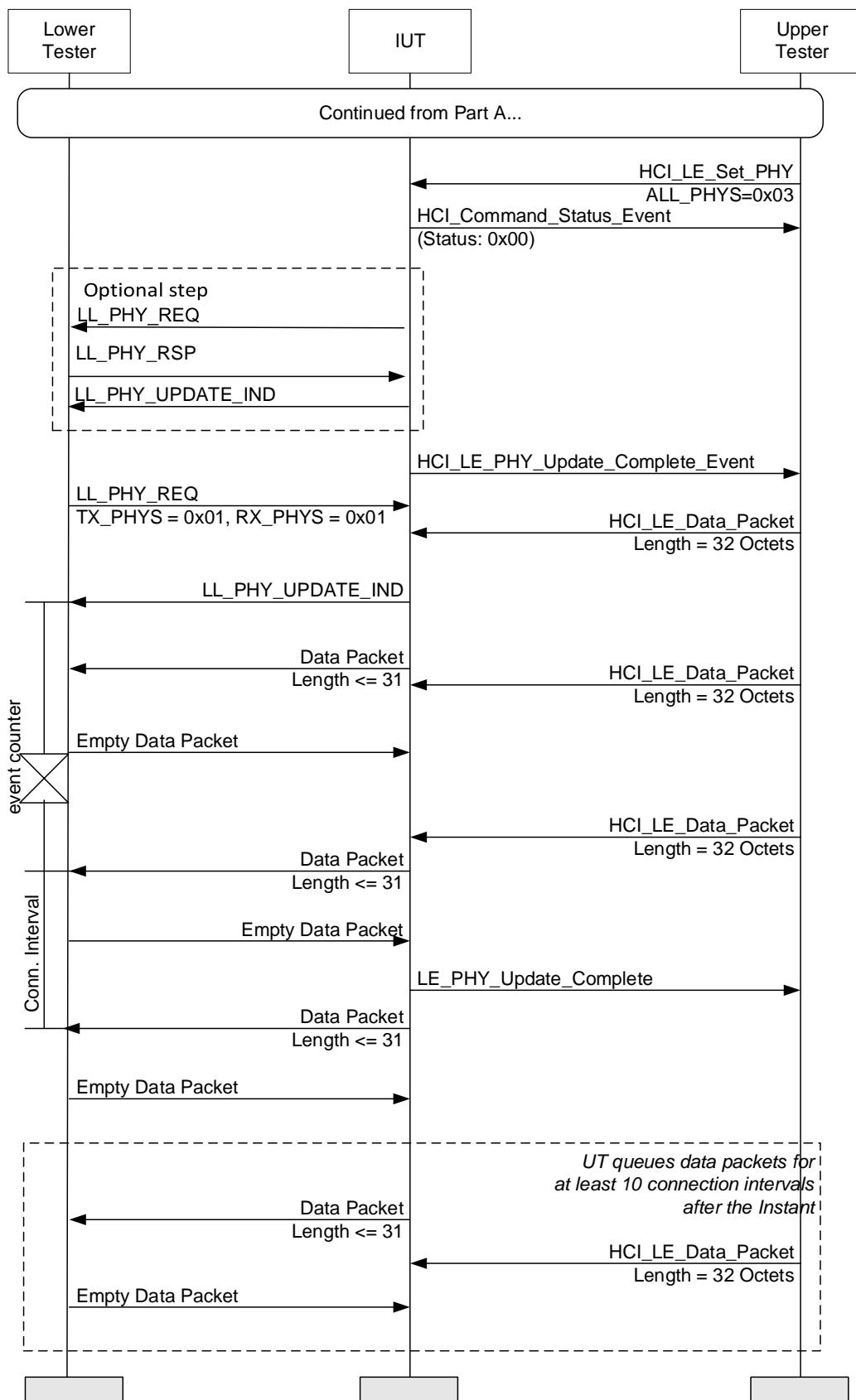


Figure 4.253: LL/CON/MAS/BV-50-C [Responding to PHY Update Procedure – Packet Time Restrictions - Part B]



1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and RX_PHYS and TX_PHYS both set to prefer the LE 2M PHY and expects an HCI_Command_Status event from the IUT in response.
2. The Lower Tester expects an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_PSP with the RX_PHYS and TX_PHYS fields both set to 0x02.
3. Lower Tester expects an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE 2M PHY for both directions. If the IUT does not select the LE 2M PHY for both directions the test case ends with an Inconclusive Verdict.
4. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE 2M PHY.
5. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x0148 and TxOctets set to 32 and expects an HCI_Command_Complete event from the IUT in response.
6. If the IUT initiates a data length update procedure, the Lower Tester responds with RxTime set to 0x0148 and RxOctets set to 32. If the IUT does not initiate a data length update procedure, the Lower Tester shall initiate the data length update procedure with RxTime set to 0x0148 and RxOctets set to 32 and expect a response from the IUT. If the IUT's TxTime < 0x0148 or TxOctets < 32 then the test case ends with an Inconclusive Verdict.
7. The Upper Tester begins to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
8. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
9. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03, and expects an HCI_Command_Status event from the IUT in response. The controller may send a LL_PHY_REQ to the Lower Tester. In this case, the Lower Tester sends an LL_PHY_RSP specifying the current PHY in both directions in response and the IUT completes the transaction with an LL_PHY_UPDATE_IND.
10. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 2M PHY.
11. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x01.
12. Lower Tester expects an LL_PHY_UPDATE_IND selecting the LE 1M PHY for both directions. If the IUT does not select the LE 1M PHY for both directions the test case ends with an Inconclusive Verdict.
13. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 12) must be 31 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY update procedure.
14. The Lower Tester and IUT complete the PHY update procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 32 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
15. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE 1M PHY.
16. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.
17. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 31 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.



- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in Step 12) must have a length of 32 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_UPDATE_IND in Step 12) must have a length of less than or equal to 31 octets.

The IUT must transition to the LE 1M PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall:

- have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection, and
- take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 2M PHY for both directions in Step 3).
- The IUT specifies values of TxTime < 0x0148 or TxOctets < 32 in Step 6).
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in Step 12).
- The IUT does not select the LE 1M PHY for both directions in Step 12).

4.3.5.42 LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]

- Test Purpose

Test that a master IUT correctly handles the case where it initiates a PHY update procedure but no common PHYs are available.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT.

- Reference

[10] 5.1.10

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

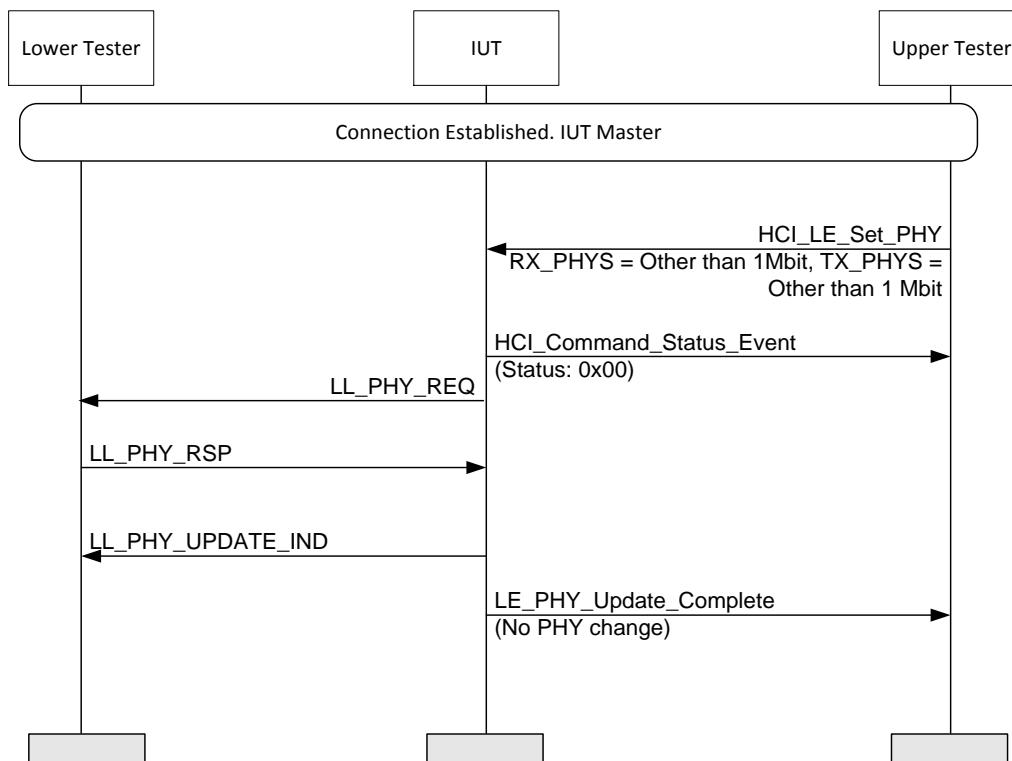


Figure 4.254: LL/CON/MAS/BV-51-C [Initiating PHY Update Procedure – No Common PHY]

- Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with **ALL_PHYS** set to zero, **RX_PHYS** and **TX_PHYS** both set to prefer a single supported PHY other than LE 1M, and **PHY_options** set to 0x0000 and expects an **HCI_Command_Status** event from the IUT in response.
- The Lower Tester expects an **LL_PHY_REQ** control PDU from the IUT with at least one bit set in each field (**RX_PHYS**, **TX_PHYS**). If the IUT does not send an **LL_PHY_REQ** the test case ends with an **Inconclusive Verdict**.
- Lower Tester acknowledges the IUT's request and responds with an **LL_PHY_RSP** PDU. For each field the Lower Tester will select a value based on the following rules:
 - If the IUT sets a single bit in a field of the **LL_PHY_REQ**, the Lower Tester will set a different bit in the corresponding field of the **LL_PHY_RSP**. The bit set must correspond to a PHY that the IUT supports.
 - If the IUT sets more than one bit in a field of the **LL_PHY_REQ**, the Lower Tester will set the bit corresponding to the current PHY.
- Lower Tester expects an **LL_PHY_UPDATE_IND** with both the **M_TO_S_PHY** and **S_TO_M_PHY** fields set to zero. The **Instant** field must also be zero.
- Upper Tester expects an **LE_PHY_Update_Complete** event from the IUT with a payload consistent with the prior PHY(s) indicating no change has occurred.

- Expected Outcome

Pass Verdict

For all cases described in the test procedure, the following conditions shall occur:

- The IUT transmits an **LL_PHY_UPDATE_IND** PDU where the **M_TO_S_PHY** and **S_TO_M_PHY** fields are both set to zero (no change). The **Instant** field is also set to zero.



- The IUT reports the prior PHY(s) with a LE_PHY_Update_Complete event indicating no change has occurred.

Inconclusive Verdict

The IUT does not send an LL_PHY_REQ in Step 2).

4.3.5.43 LL/CON/MAS/BV-52-C [Master Receiving Data, LE Coded, CI Change]

- Test Purpose

Test that a master IUT is able to receive data from a slave device when the slave is transitioning between 125kbit and 500kbit coded rates. Confirm that IUT responds within the allowed TIFS times for each packet at either coded rate. Test is performed with the IUT's minimum and maximum supported packet length. A Data Length Update Procedure is performed if required.

The Lower Tester acts in the slave role in the connection, sends data to the IUT according to the acknowledgement scheme and observes the data reported to the host of the IUT.

- Reference

[10] 4.5

- Initial Condition

Parameters: LL_master_payload_length_MIN, LL_master_payload_length_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, maximum supported connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure using the connection handle and data packet length from the execution of the preamble steps.



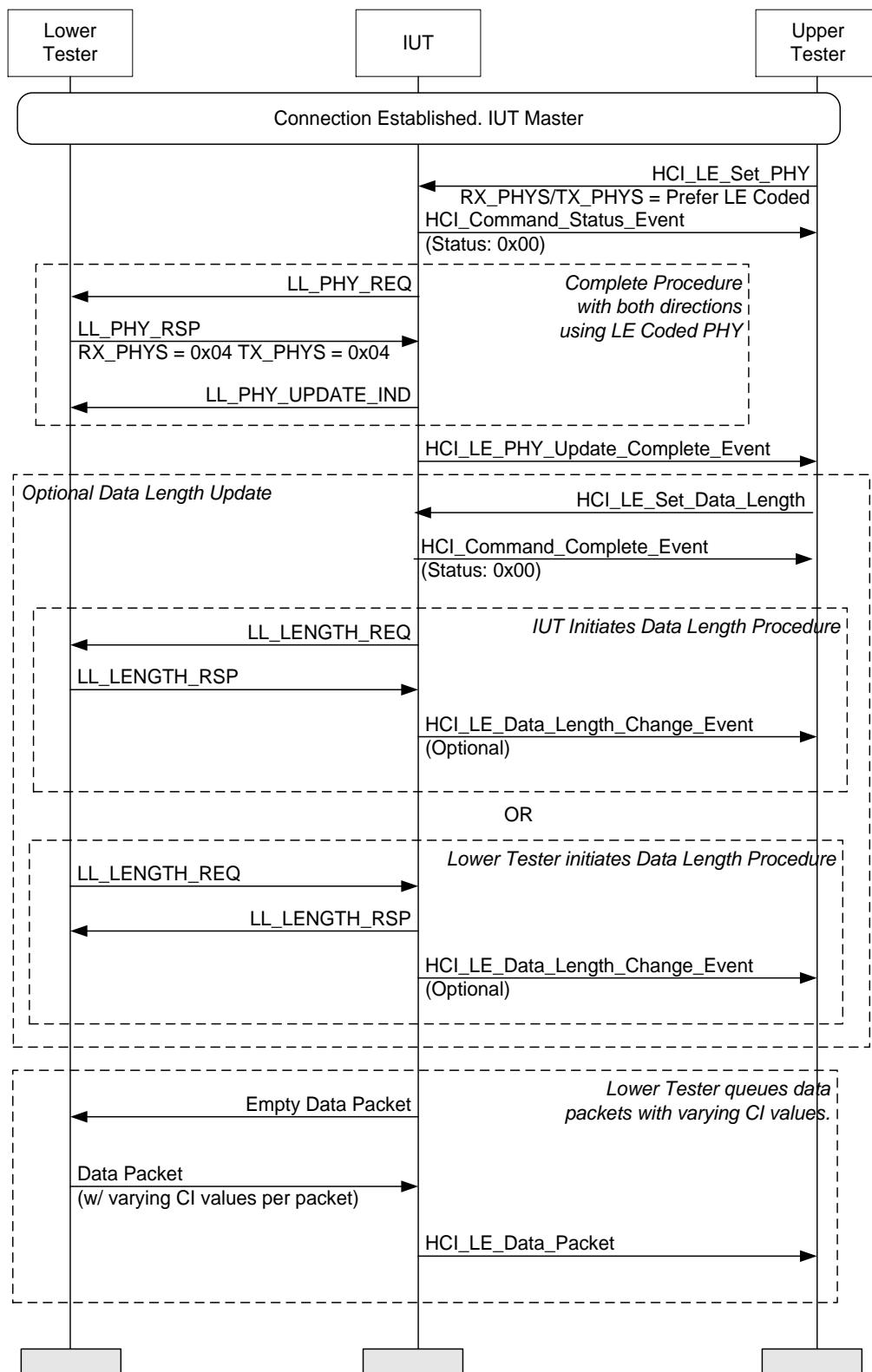


Figure 4.255: LL/CON/MAS/BV-52-C [Master Receiving Data, LE Coded, CI Change]

1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT ALL_PHYS set to zero, RX_PHYS and TX_PHYS both set to prefer LE Coded PHY, and PHY_options set to 0x0000, and expects an **HCI_Command_Status_event** from the IUT in response.

2. The Lower Tester expects an LL_PHY_REQ PDU from the IUT and responds with an LL_PHY_PSP with the RX_PHYS and TX_PHYS fields both set to prefer the LE Coded PHY.
3. Lower Tester expects an LL_PHY_UPDATE_IND PDU from the IUT selecting the LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
4. The Upper Tester expects an HCI_LE_PHY_Update_Complete event indicating both directions are operating using the LE Coded PHY.
5. If (TSPX_TxOctets_Max > 27) OR (TSPX_TxTime_Max > 328) then the Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to TSPX_TxTime_Max and TxOctets set to TSPX_TxOctets_Max and expects an HCI_Command_Complete event from the IUT in response, otherwise go to step 7.
6. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure TxTime set to TSPX_RxTime_Max and TxOctets set to TSPX_RxOctets_Max and expects a response from the IUT.
7. Configure Lower Tester to send 288 data packets of length connEffectiveMaxTxOctets with all payload octets set to continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFE, 0xFF, 0x00, 0x01, etc.). For each packet sent the CI field will alternate in the pattern below. Each number in the pattern represents the CI field for that particular packet to be sent.

```

2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8
2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8
2 2 2 2 2 2 2 2 8 2 2 2 2 2 2 2
2 8 8 8 8 8 8 8 2 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 8 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 2 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 2 2 2
8 8 8 2 2 2 8 8 8 2 8 8 2 2 2 8
2 8 2 2 8 2 8 8 8 8 8 2 8 2 8 2
8 2 2 2 2 8 2 8 8 2 8 8 8 8 2 2
8 8 8 2 2 8 2 8 2 8 8 2 2 8 8 2
2 2 2 2 8 2 8 8 2 8 2 8 8 8 2
8 2 2 2 8 2 2 8 2 2 2 8 2 2 2
2 2 2 8 2 2 8 2 2 8 8 2 8 2 2 8
8 8 8 2 8 8 8 2 2 2 2 8 8 8 8 8

```

8. Lower Tester sends one or more DATA packets to the IUT, using the acknowledgement scheme and the data channel selection parameters. Lower Tester expects an empty DATA packet in



- response from the IUT for each DATA packet sent by the Lower Tester, received with acceptable TIFS timing (150 µs +/- 2 µs).
9. Upper Tester expects an HCI_LE_Data_Packet event from the IUT containing one or more data elements sent in step 8. The received data should be in the form of continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFE, 0xFF, 0x00, 0x01, etc.). Repeat step 7 through 9 until all data to be sent has been reported to the Upper Tester.
 10. Repeat the entire data exchange procedure again from Step 7 through 10, but utilizing the minimum data payload size (27 bytes) instead of connEffectiveMaxTxOctets for each data packet.
- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent.

The IUT reports all data correctly with HCI_Data_Packet events containing continuously incrementing values, starting with 0x00 in the first data packet (0x00, 0x01, 0x02, 0x03 ... 0xFE, 0xFF, 0x00, 0x01, etc.).

All responses from the IUT are received with acceptable TIFS timing (150 µs +/- 2 µs)

4.3.5.44 LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded]

- Test Purpose

Test that a master IUT follows all packet time restrictions both during and after PHY update when it initiates the PHY Update Procedure. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester acts in the slave role maintaining a connection. A Data Length Update Procedure is performed. The Upper Tester begins queuing data to the IUT and issues the HCI command to start the PHY Update Procedure as the Host of the IUT. The Lower Tester observes the PHY Update Procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the procedure.

- Reference
- [10] 5.1.10.1
- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

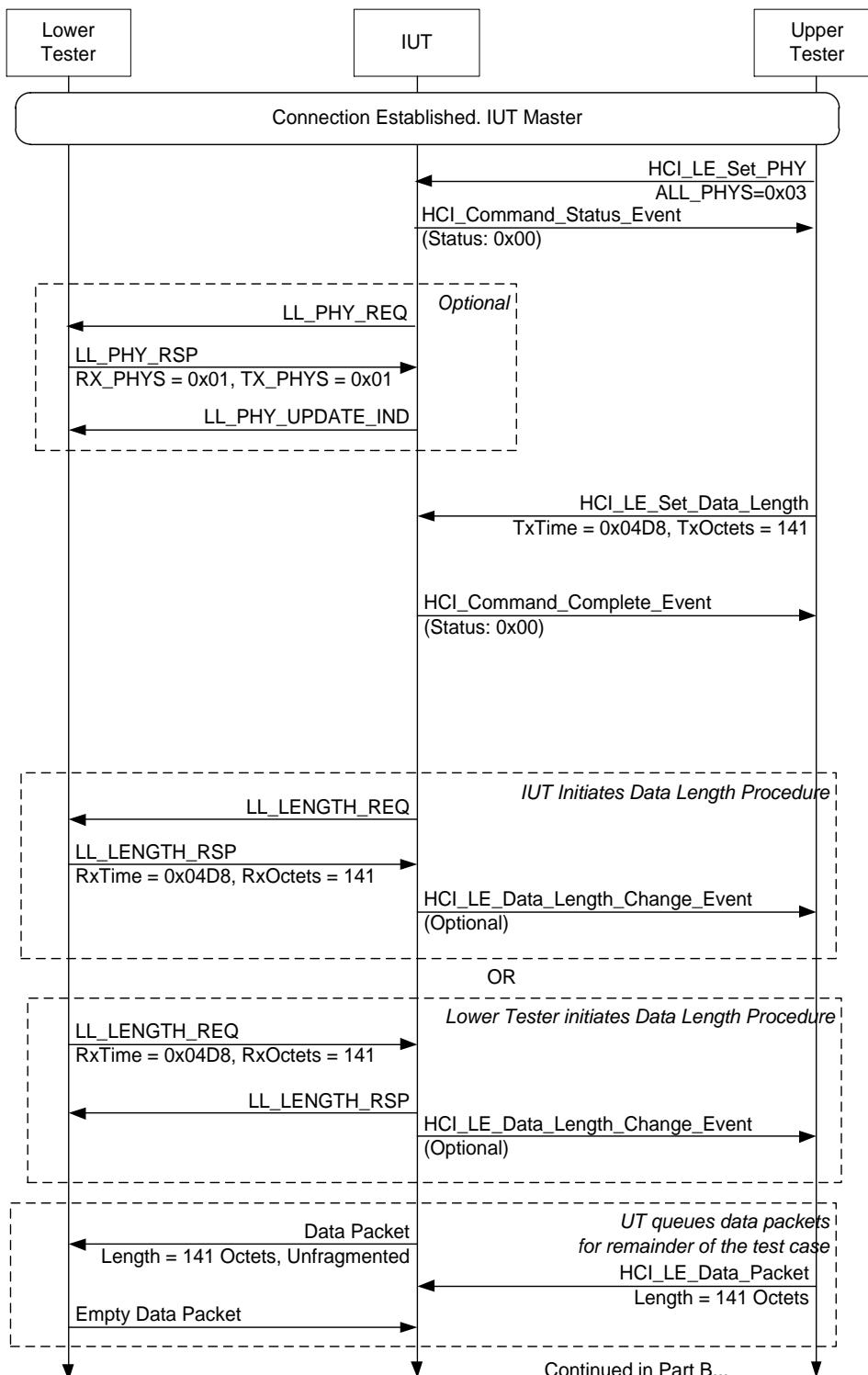


Figure 4.256: LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part A]



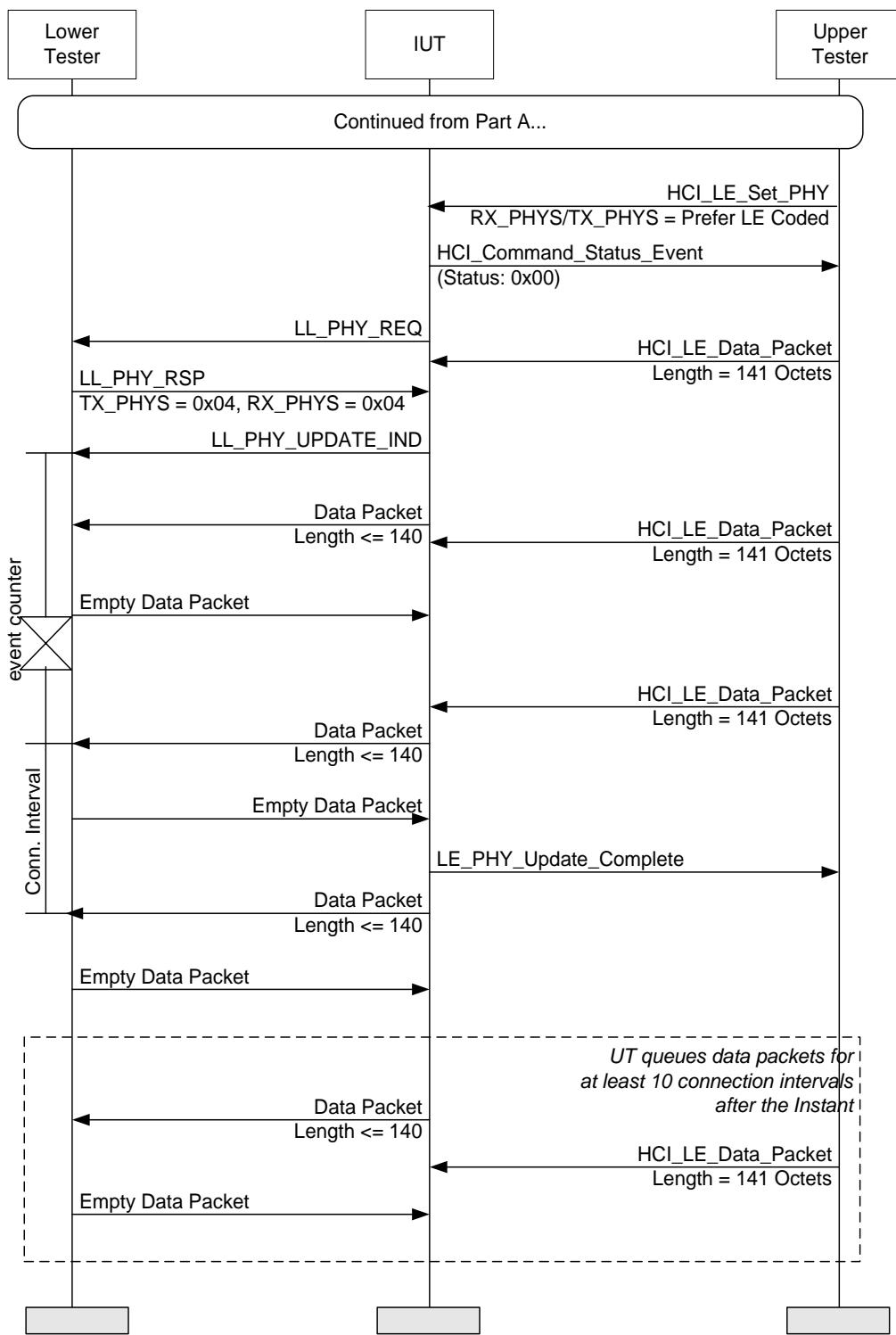


Figure 4.257: LL/CON/MAS/BV-53-C [Initiating PHY Update Procedure – Packet Time Restrictions, LE Coded – Part B]

1. Upper Tester sends an **HCI_LE_Set_PHY** command to the IUT with **ALL_PHYS** set to 0x03 and **PHY_options** set to 0x0000, and expects an **HCI_Command_Status** event from the IUT in response.
2. The Upper Tester expects an **HCI_LE_PHY_Update_Complete** event from the IUT indicating both directions are operating using the LE 1M PHY.

3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
 4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
 5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with an Inconclusive Verdict.
 7. The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero, PHY_options set to 0x0000, and RX_PHYS and TX_PHYS both set to prefer LE Coded PHY.
 8. The Lower Tester expects an LL_PHY_REQ from the IUT and responds with an LL_PHY_RSP with the RX_PHYS and TX_PHYS fields both set to 0x04.
 9. The Lower Tester expects an LL_PHY_UPDATE_IND from the IUT selecting the LE Coded PHY to be used in both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
 10. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 13 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY Update Procedure.
 11. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 12. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.
 13. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.
 14. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in Step 9 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_UPDATE_IND in Step 9 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and



connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.

Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 1M PHY for both directions in Step 2.
- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in Step 9.
- The IUT does not select the LE Coded PHY for both directions in Step 9.

4.3.5.45 LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded]

- **Test Purpose**

Test that a master IUT both during and after PHY update when it responds to a PHY Update Procedure from a slave device. In particular, test that the IUT does not queue a packet for transmission that would satisfy the requirements when queued but violate them if it is still waiting for retransmission after the PHY Update instant.

The Lower Tester acts in the slave role maintaining a connection. A Data Length Update Procedure is performed. The Upper Tester begins queueing data to the IUT and starts the PHY Update Procedure. The Lower Tester observes the PHY Update Procedure carried out by the IUT and ensures that all packet time restrictions are followed both before and after the change.

- **Reference**

[10] 5.1.10.1

- **Initial Condition**

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

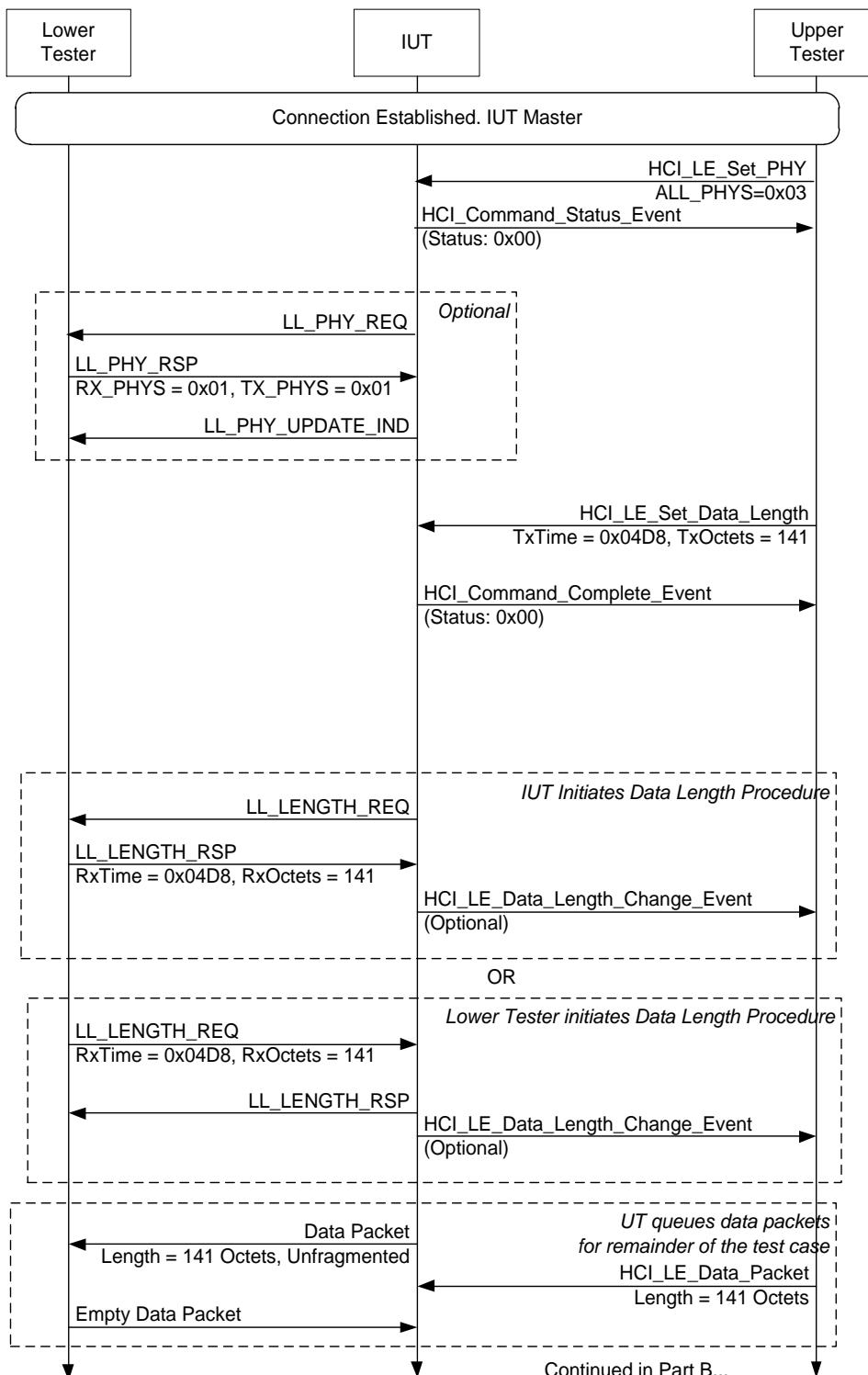


Figure 4.258: LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LECoded – Part A]



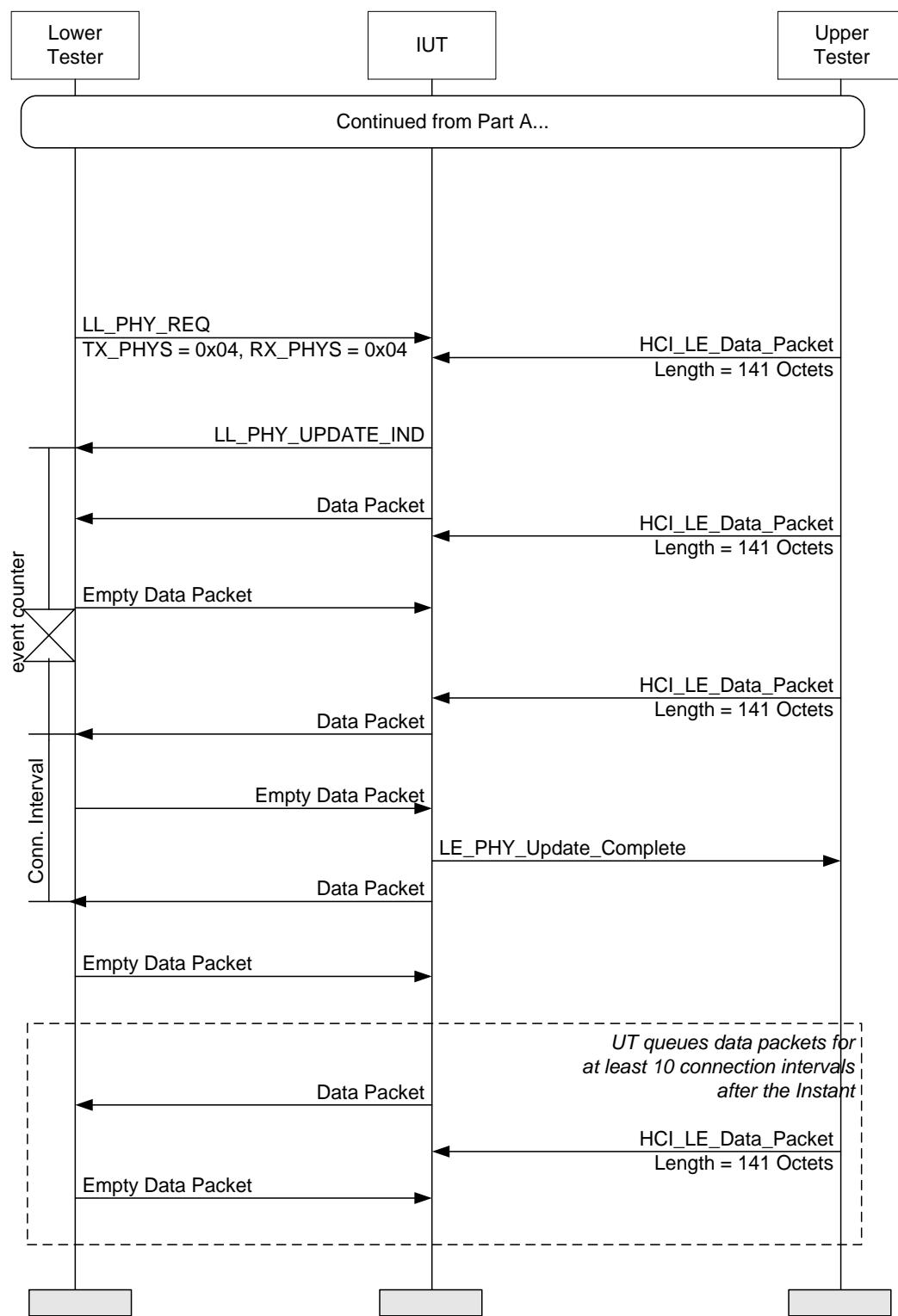


Figure 4.259: LL/CON/MAS/BV-54-C [Responding to PHY Update Procedure – Packet Time Restrictions, LE Coded - Part B]

1. Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x03 and PHY_options set to 0x0000, and expects an HCI_Command_Status event from the IUT in response.
2. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE 1M PHY.



3. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT with TxTime set to 0x04D8 and TxOctets set to 141 and expects an HCI_Command_Complete event from the IUT in response.
 4. If the IUT initiates a Data Length Update Procedure, the Lower Tester responds with RxTime set to 0x04D8 and RxOctets set to 141. If the IUT does not initiate a Data Length Update Procedure, the Lower Tester shall initiate the Data Length Update Procedure with RxTime set to 0x04D8 and RxOctets set to 141 and expect a response from the IUT. If the IUT's TxTime < 0x04D8 or TxOctets < 141 then the test case ends with an Inconclusive Verdict.
 5. The Upper Tester begins to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start). The Upper Tester continues to queue additional data packets throughout the remainder of the test case.
 6. The Lower Tester expects to receive un-fragmented data packets matching those queued by the Upper Tester. If the packets are fragmented the test case ends with a fail verdict.
 7. The Lower Tester sends an LL_PHY_REQ PDU to the IUT with the RX_PHYS and TX_PHYS field both set to 0x04.
 8. Lower Tester expects an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.
 9. The Lower Tester expects that all data packets received after the IUT sends the LL_PHY_UPDATE_IND in step 13 must be 140 octets or less in length. If a larger packet is received the test case ends with a failed verdict. The Lower Tester expects to receive data packets both before and after the PHY Update Procedure.
 10. The Lower Tester and IUT complete the PHY Update Procedure with the Upper Tester continuing to queue data packets to the IUT with a length of 141 octets and the LLID set to 10b (start) and the Lower Tester receiving data packets.
 11. At the Instant specified in the LL_PHY_UPDATE_IND the IUT begins to maintain the connection using the LE Coded PHY.
 12. The Upper Tester expects an HCI_LE_PHY_Update_Complete event from the IUT indicating both directions are operating using the LE Coded PHY.
 13. The Upper Tester continues to queue data packets and the Lower Tester continues to receive data packets and confirm that they are 140 octets or less in length. This data exchange continues for at least 10 connection events after the Instant.
- Expected Outcome

Pass Verdict

All data packets received by the Lower Tester before the IUT sends an LL_PHY_UPDATE_IND in Step 8 must have a length of 141 octets and be un-fragmented.

All data packets received by the Lower Tester after the IUT sends an LL_PHY_UPDATE_IND in Step 8 must have a length of less than or equal to 140 octets.

The IUT must transition to the LE Coded PHY at the Instant specified in the LL_PHY_UPDATE_IND PDU.

All data packets received by the Lower Tester shall have a Payload Length less than or equal to the current value of connEffectiveMaxRxOctets for the connection and take a total time to receive less than or equal to the current value of connEffectiveMaxRxTime for the connection.

For the first two data packets received (including any retransmissions of those packets) after each Data Length Update Procedure is completed, the values of connEffectiveMaxRxOctets and connEffectiveMaxRxTime that apply shall be the greater of those in effect before and after the procedure was carried out.



Inconclusive Verdict

One or more of the following:

- The IUT does not select the LE 1M PHY for both directions in Step 2.
- The IUT specifies values of TxTime < 0x04D8 or TxOctets < 141 for the Data Length Update Procedure in Step 4.
- The IUT sends fragmented packets before the IUT sends an LL_PHY_UPDATE_IND in Step 8.

The IUT does not select the LE Coded PHY for both directions in Step 8.

4.3.5.46 LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded]

- **Test Purpose**

Test that a master IUT still transmits data even when the TxTime and/or RxTime values for LE Coded PHY suggest a smaller possible data length than the minimum length data PDU (27 octets).

The Lower Tester acts in the slave role maintaining a connection. If a Data Length Update Procedure is performed, minimum settings are used. The Upper Tester begins queuing data to the IUT. The Lower Tester then initiates the PHY Update Procedure. Lower Tester observes the procedure carried out by the IUT and ensures PDUs are still transmitted after the procedure.

- **Reference**

[10] 5.1.9

- **Initial Condition**

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

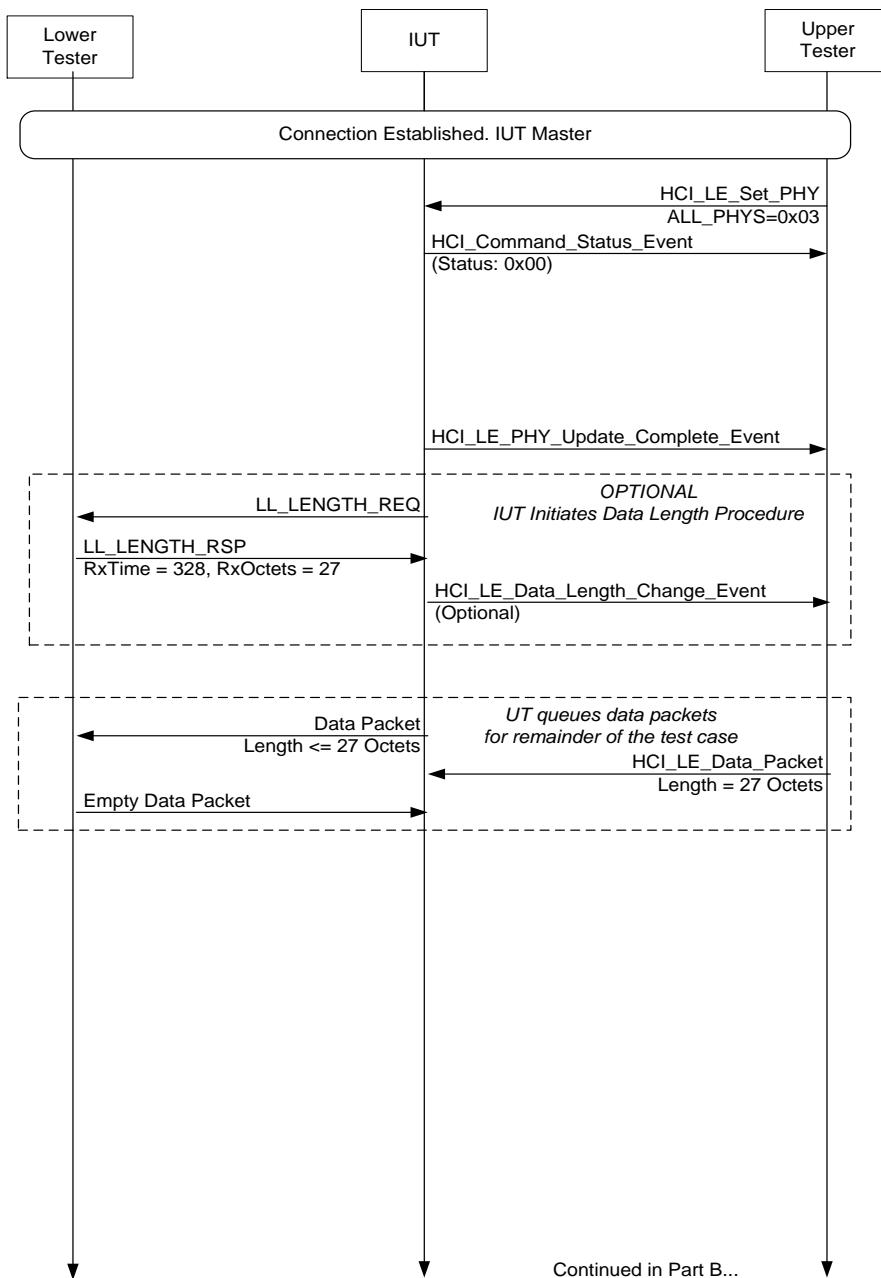


Figure 4.260: LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded – Part A]



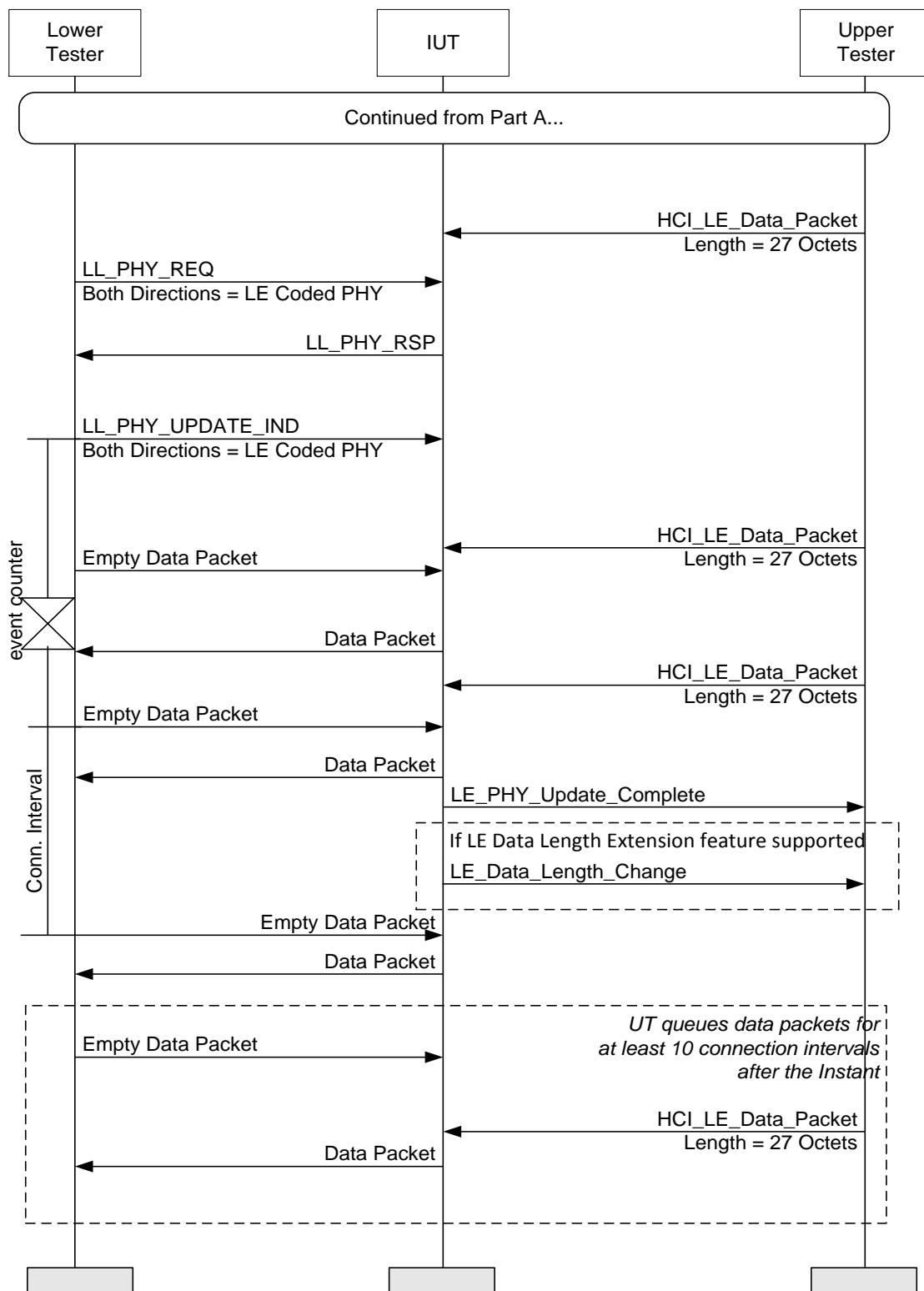


Figure 4.261: LL/CON/MAS/BV-55-C [Mandatory Minimum PDU Length, LE Coded - Part B]

The test procedure and Expected Outcome are the same as test LL/CON/SLA/BV-57-C except that step 6 is:

6. Lower Tester expects an LL_PHY_UPDATE_IND selecting LE Coded PHY for both directions. If the IUT does not select the LE Coded PHY for both directions the test case ends with an Inconclusive Verdict.

4.3.5.47 LL/CON/MAS/BV-73-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY]

- **Test Purpose**

Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE 1M PHY.

- **Reference**

[8] 5.1.9, [10] 4.5.10

- **Initial Condition**

State: Connected Master. Values for maximum TxOctets supported ($\text{supportedMaxTxOctets}_{\text{IUT}}$), TxTime ($\text{supportedMaxTxTime}_{\text{IUT}}$), RxOctets ($\text{supportedMaxRxOctets}_{\text{IUT}}$) and RxTime ($\text{supportedMaxRxTime}_{\text{IUT}}$) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in “Octet”
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in “Octet”
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in “Time”
- connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in “Time”

The values of these parameters will depend on the features supported by the IUT.



- Test Procedure

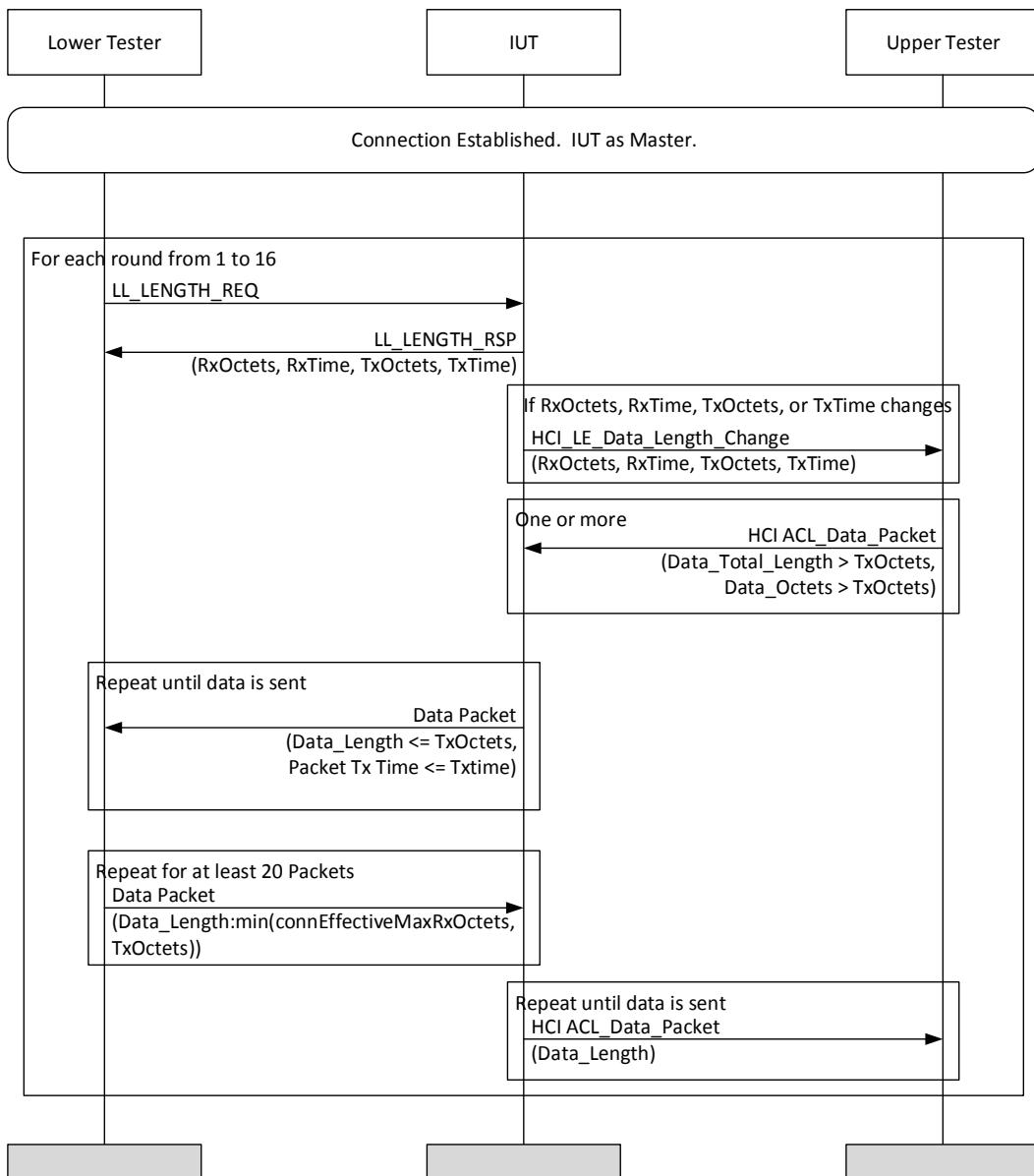


Figure 4.262: LL/CON/MAS/BV-73-C: Master Data Length Update – Responding to Data Length Update Procedure

- For each round in [Table 4.43](#), perform steps 2 to 9. Note: if two or more rounds turn out to use the same values, all except the first may be omitted.
- The Lower Tester sends an LL_LENGTH_REQ to the IUT with the payload specified for the relevant round.
- The Lower Tester expects an LL_LENGTH_RSP from the IUT with a value selected for
 - connMaxTxOctetsIUT between connMinOctetsLimitSpec and supportedMaxTxOctetsIUT
 - connMaxTxTimeIUT between connMinTimeLimitSpec and supportedMaxTxTimeIUT
 - connMaxRxOctetsIUT between connMinOctetsLimitSpec and supportedMaxRxOctetsIUT
 - connMaxRxTimeIUT between connMinTimeLimitSpec and supportedMaxRxTimeIUT



4. If the values in either the LL_LENGTH_REQ or LL_LENGTH_RSP PDUs mean the values of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime have changed, the Upper Tester expects an HCI_LE_Data_Length_Change event from the IUT containing the new values. Otherwise it expects not to receive such an event. Note: in the very first round it might not be possible to determine whether the values have changed.
5. The Upper Tester sends one or more HCI ACL DATA packets to the IUT with the total Data Length greater than supportedMaxTxOctets IUT and greater than the maximum number of data octets that the IUT can send in a packet taking no more than supportedMaxTxTime IUT to transmit on the current PHY.
6. The Lower Tester expects to receive DATA packets until all the data has been transmitted, with each packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no longer than connEffectiveMaxTxTime microseconds to transmit.
7. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester in step 5.
8. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to smaller of the IUT's connEffectiveMaxRxOctets and the maximum number of octets that can be included in a packet that can be transmitted on the current PHY in no longer than the IUT's connEffectiveMaxRxTime microseconds.
9. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects the data transmitted by the IUT to match the data sent by the Lower Tester in step 8.
10. If the IUT supports the LE Coded PHY, then repeat steps 2 to 9 but only performing the rounds where either connMaxTxTimeTester or connMaxRxTimeTester is connMaxTimeLimitSpecs and, in those rounds, replacing connMaxTimeLimitSpec by 2120 µs.

Round	LL_LENGTH_REQ			
	connMaxTxOctetsTester	connMaxTxTimeTester	connMaxRxOctetsTester	connMaxRxTimeTester
1	connMinOctetsLimitSpec	connMinTimeLimitSpec	connMinOctetsLimitSpec	connMinTimeLimitSpec
2	connMaxOctetsLimitSpec	connMaxTimeLimitSpec	connMaxOctetsLimitSpec	connMaxTimeLimitSpec
3	supportedMaxRxOctetsIUT	supportedMaxRxTimeIUT	supportedMaxTxOctetsIUT	supportedMaxTxTimeIUT
4	connMinOctetsLimitSpec	connMinTimeLimitSpec	supportedMaxTxOctetsIUT	supportedMaxTxTimeIUT
5	connMinOctetsLimitSpec	connMaxTimeLimitSpec	connMinOctetsLimitSpec	connMaxTimeLimitSpec
6	connMinOctetsLimitSpec	supportedMaxRxTimeIUT	connMaxOctetsLimitSpec	connMaxTimeLimitSpec
7	connMaxOctetsLimitSpec	connMinTimeLimitSpec	supportedMaxTxOctetsIUT	connMaxTimeLimitSpec
8	connMaxOctetsLimitSpec	connMaxTimeLimitSpec	connMinOctetsLimitSpec	connMinTimeLimitSpec
9	connMaxOctetsLimitSpec	supportedMaxRxTimeIUT	connMaxOctetsLimitSpec	connMinTimeLimitSpec
10	supportedMaxRxOctetsIUT	connMinTimeLimitSpec	supportedMaxTxOctetsIUT	connMinTimeLimitSpec
11	supportedMaxRxOctetsIUT	connMaxTimeLimitSpec	connMinOctetsLimitSpec	supportedMaxTxTimeIUT
12	supportedMaxRxOctetsIUT	supportedMaxRxTimeIUT	connMaxOctetsLimitSpec	supportedMaxTxTimeIUT
13-16	The connMaxTxOctetsTester and connMaxRxOctetsTester fields shall each be set to a randomly-selected value greater than connMinOctetsLimitSpec, less than connMaxOctetsLimitSpec, and not equal to either supportedMaxTxOctetsTester or supportedMaxRxOctetsTester. The connMaxTxTimeTester and connMaxRxTimeTester fields shall each be set to a randomly-selected value greater than connMinTimeLimitSpec, less than connMaxTimeLimitSpec, and not equal to either supportedMaxTxTimeTester or supportedMaxRxTimeTester. If no such value exists for a parameter then use the value from round 1 instead.			

Table 4.43: LL_LENGTH_REQ content



- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT responding to the Lower Tester's LL_LENGTH_REQ PDU with an LL_LENGTH_RSP PDU.

The IUT sends an HCI_LE_Data_Length_Change event if at least one of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime has changed since the previous round and the event contains the correct values. The IUT does not send an event if none of the values have changed.

The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations.

4.3.5.48 LL/CON/MAS/BV-74-C [Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY]

- Test Purpose

Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE 1M PHY.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the Data Length update procedure as the Host of the IUT, and the Lower Tester then observes the Data Length update procedure carried out by the IUT and accepts the IUT's request.

- Reference

[8] 5.1.9

[10] [8] 4.5.10

State: Connected Master. Values for maximum TxOctets supported (supportedMaxTxOctets_{IUT}), TxTime (supportedMaxTxTime_{IUT}), RxOctets (supportedMaxRxOctets_{IUT}) and RxTime (supportedMaxRxTime_{IUT}) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in "Octet"
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in "Octet"
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in "Time"
- connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in "Time"

The values of these parameters will depend on the features supported by the IUT.



- Test Procedure

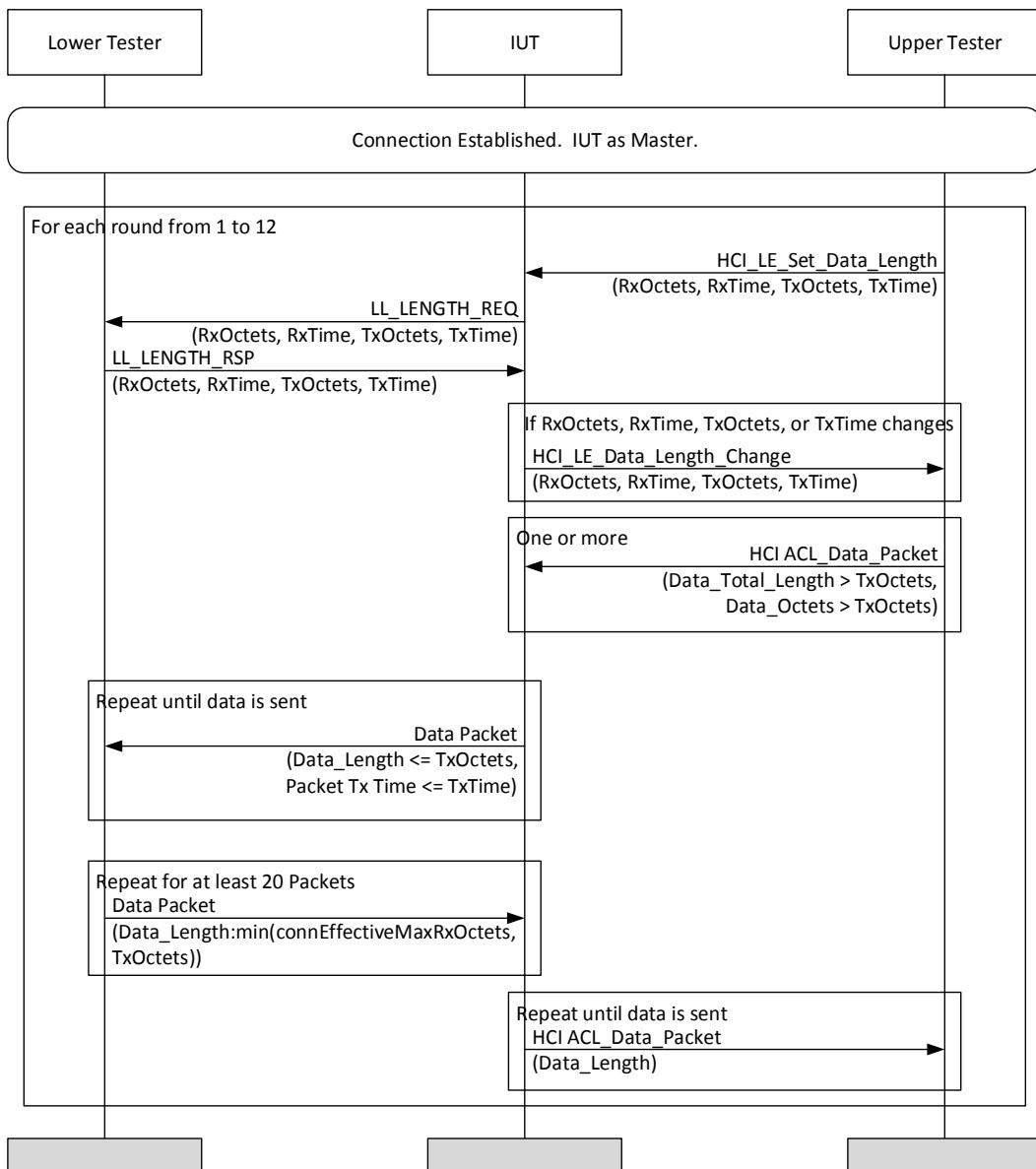


Figure 4.263: LL/CON/MAS/BV-74-C: Master Data Length Update – Responding to Data Length Update Procedure.

- For each round in [Table 4.44](#), perform steps 2 to 10. Note: if two or more rounds turn out to use the same values, all except the first may be omitted; therefore rounds 10-12 are only performed if the IUT supports the LE Coded PHY.
- The Upper Tester sends an **HCI LE Set Data Length** command to the IUT with the payload specified for the relevant round.
- The Lower Tester expects an **LL LENGTH REQ** from the IUT with a value selected for
 - $\text{connMaxTxOctets}_{\text{IUT}}$ between $\text{connMinOctetsLimitSpec}$ and $\text{supportedMaxTxOctets}_{\text{IUT}}$
 - $\text{connMaxTxTime}_{\text{IUT}}$ between $\text{connMinTimeLimitSpec}$ and $\text{supportedMaxTxTime}_{\text{IUT}}$
 - $\text{connMaxRxOctets}_{\text{IUT}}$ between $\text{connMinOctetsLimitSpec}$ and $\text{supportedMaxRxOctets}_{\text{IUT}}$



- connMaxRxTime_{IUT} between connMinTimeLimitSpec and supportedMaxRxTime_{IUT}

If no such PDU is sent, skip to the start of the next round.

4. The Lower Tester responds with an LL_LENGTH_RSP PDU with the two octet fields set to connMinOctetsLimitSpec and the two time fields set to connMinTimeLimitSpec.
5. If the values in either the LL_LENGTH_REQ or LL_LENGTH_RSP PDUs mean the values of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime have changed, the Upper Tester expects an HCI_LE_Data_Length_Change event from the IUT containing the new values. Otherwise it expects not to receive such an event. Note: in the very first round it might not be possible to determine whether the values have changed. Note: the IUT might send two events if it makes an autonomous change and then the Lower Tester's response also causes an event.
6. The Upper Tester sends one or more HCI ACL DATA packets to the IUT with the total Data Length greater than supportedMaxTxOctets IUT and greater than the maximum number of data octets that the IUT can send in a packet taking no more than supportedMaxTxTime IUT to transmit on the current PHY.
7. The Lower Tester expects to receive DATA packets until all the data has been transmitted, with each packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no longer than connEffectiveMaxTxTime microseconds to transmit.
8. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester in step 6.
9. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to smaller of the IUT's connEffectiveMaxRxOctets and the maximum number of octets that can be included in a packet that can be transmitted on the current PHY in no longer than the IUT's connEffectiveMaxRxTime microseconds.
10. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects the data transmitted by the IUT to match the data sent by the Lower Tester in step 9.
11. Repeat steps 1 to 10 but, in step 4, set the two octet fields to connMaxOctetsLimitSpec and the two time fields to connMaxTimeLimitSpec.

Round	HCI LE Set Data Length	
	TxOctets	TxTime
1	supportedMaxTxOctetsIUT	supportedMaxTxTimeIUT
2	connMinOctetsLimitSpecs	supportedMaxTxTimeIUT
3	connMaxOctetsLimitSpecs	supportedMaxTxTimeIUT
4	supportedMaxTxOctetsIUT	connMinTimeLimitSpecs
5	connMinOctetsLimitSpecs	connMinTimeLimitSpecs
6	connMaxOctetsLimitSpecs	connMinTimeLimitSpecs
7	supportedMaxTxOctetsIUT	connMaxTimeLimitSpec
8	connMinOctetsLimitSpecs	connMaxTimeLimitSpec
9	connMaxOctetsLimitSpecs	connMaxTimeLimitSpec
10	supportedMaxTxOctetsIUT	2120 µs
11	connMinOctetsLimitSpecs	2120 µs
12	connMaxOctetsLimitSpecs	2120 µs

Table 4.44: HCI LE Set Data Length content



- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT transmitting an LL_LENGTH_REQ PDU to the Lower Tester.

The IUT sends an HCI_LE_Data_Length_Change event if at least one of connEffectiveMaxTxOctets, connEffectiveMaxRxOctets, connEffectiveMaxTxTime, or connEffectiveMaxRxTime has changed since the previous round and the event contains the correct values. The IUT does not send an event if none of the values have changed.

The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations.

Inconclusive Verdict

The IUT does not initiate at least one Data Length Update Procedure by sending an LL_LENGTH_REQ to the Lower Tester during this test case.

4.3.5.49 LL/CON/MAS/BV-75-C [Master Data Length Update – Slave does not support; LE 1M PHY]

- Test Purpose

Verify that the IUT as Master correctly handles communication with a Lower Tester that does not support the Data Length Update Procedure.

- Reference

[8] 5.1.9

- Initial Condition

State: Connected Master. Values for maximum TxOctets supported (supportedMaxTxOctets_{IUT}), TxTime (supportedMaxTxTime_{IUT}), RxOctets (supportedMaxRxOctets_{IUT}) and RxTime (supportedMaxRxTime_{IUT}) have been declared by the manufacturer via IXIT and are within the ranges specified in the Core Specification.

Note: in this test, the following terms are used to refer to the minima and maxima in Volume 6 Part B Table 4.3 of the Core Specification:

- connMinOctetsLimitSpec = the minimum permitted value for parameters with names ending in "Octet"
- connMaxOctetsLimitSpec = the maximum permitted value for parameters with names ending in "Octet"
- connMinTimeLimitSpec = the minimum permitted value for parameters with names ending in "Time"
- connMaxTimeLimitSpec = the maximum permitted value for parameters with names ending in "Time"

The values of these parameters will depend on the features supported by the IUT.



- Test Procedure

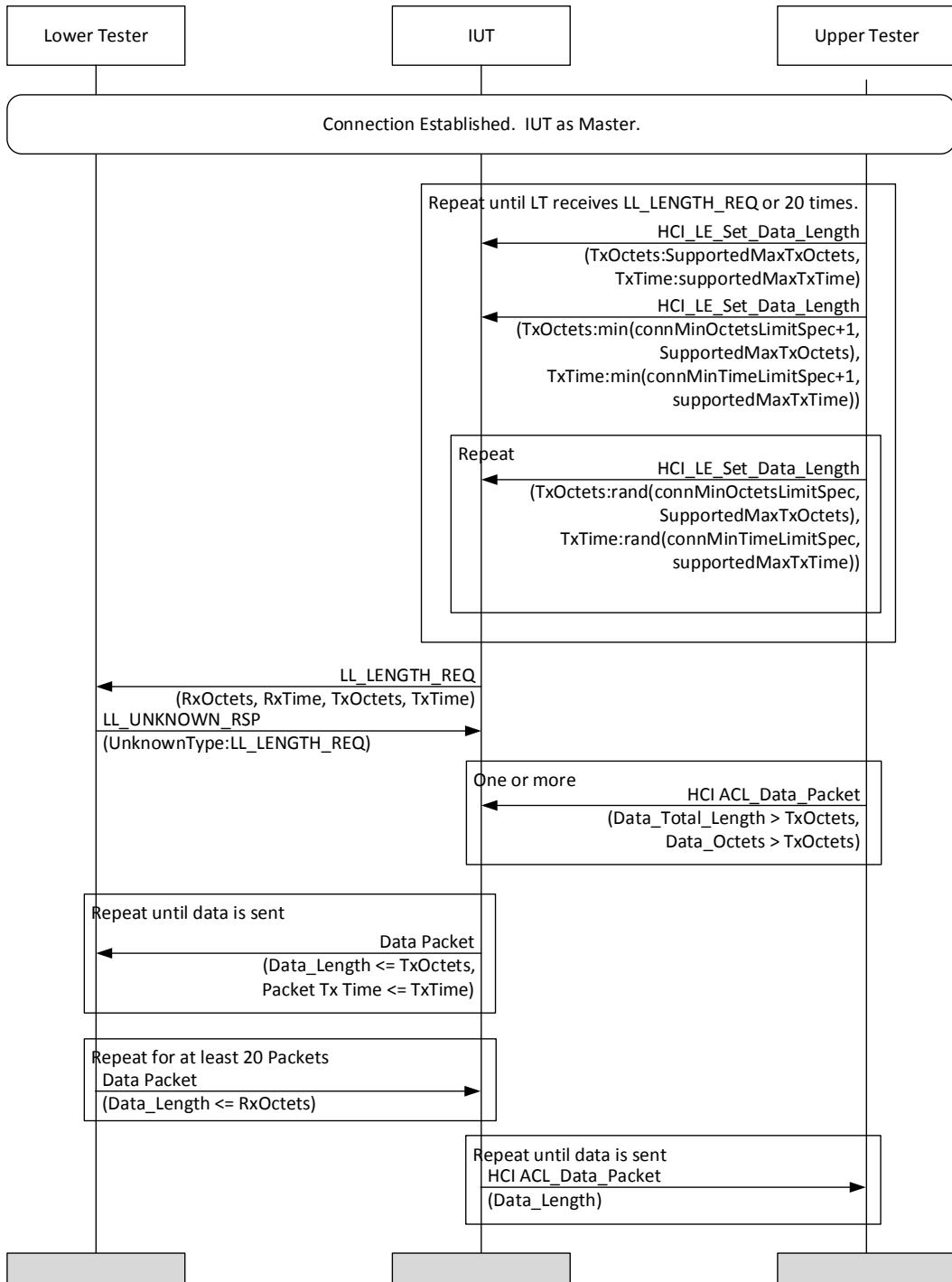


Figure 4.264: LL/CON/MAS/BV-75-C: Master Data Length Update – Responding to Data Length Update Procedure

1. Repeat step 2 until the Lower Tester receives an LL_LENGTH_REQ from the PDU or until it has been repeated 20 times. In the latter case, return an Inconclusive Verdict.
2. The Upper Tester sends an HCI_LE_Set_Data_Length command to the IUT.

- a. The first time, the TxOctets field shall be set to supportedMaxTxOctets_{IUT} and the TxTime field to supportedMaxTxTime_{IUT}.
- b. The second time, the TxOctets field shall be set to the lesser of connMinOctetsLimitSpec + 1 and supportedMaxTxOctets_{IUT} and the TxTime field to the lesser of connMinTimeLimitSpec + 1 and supportedMaxTxTime_{IUT}.
- c. The remaining times, the TxOctets field shall be set to a random value between connMinOctetsLimitSpec and supportedMaxTxOctets_{IUT} inclusive and the TxTime field to a random value between connMinTimeLimitSpec and supportedMaxTxTime_{IUT} inclusive.
- 3. The Lower Tester checks that the LL_LENGTH_REQ PDU received from the IUT has a value selected for
 - connMaxTxOctets_{IUT} between connMinOctetsLimitSpec and supportedMaxTxOctets_{IUT}
 - connMaxTxTime_{IUT} between connMinTimeLimitSpec and supportedMaxTxTime_{IUT}
 - connMaxRxOctets_{IUT} between connMinOctetsLimitSpec and supportedMaxRxOctets_{IUT}
 - connMaxRxTime_{IUT} between connMinTimeLimitSpec and supportedMaxRxTime_{IUT}
- 4. The Lower Tester responds with an LL_UNKNOWN_RSP PDU with the Unknown type set to LL_LENGTH_REQ.
- 5. The Upper Tester sends one or more HCI ACL DATA packets to the IUT with the total Data Length greater than supportedMaxTxOctets IUT and greater than the maximum number of data octets that the IUT can send in a packet taking no more than supportedMaxTxTime IUT to transmit on the current PHY.
- 6. The Lower Tester expects to receive DATA packets until all the data has been transmitted, with each packet having a payload length less than or equal to connEffectiveMaxTxOctets and taking no longer than connEffectiveMaxTxTime microseconds to transmit.
- 7. The data transmitted by the IUT to the Lower Tester matches the data sent by the Upper Tester in step 5.
- 8. The Lower Tester sends at least 20 DATA packets to the IUT with the payload length equal to smaller of the IUT's connEffectiveMaxRxOctets and the maximum number of octets that can be included in a packet that can be transmitted on the current PHY in no longer than the IUT's connEffectiveMaxRxTime microseconds.
- 9. The IUT sends one or more HCI ACL DATA packets to the Upper Tester. Upper Tester expects the data transmitted by the IUT to match the data sent by the Lower Tester in step 8.

- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the IUT transmitting an LL_LENGTH_REQ PDU to the Lower Tester.

The IUT transmits DATA packets to the Upper and Lower Testers respecting Data Length limitations

Fail Verdict

The IUT sends an HCI_LE_Data_Length_Change event to the Upper Tester.

Inconclusive Verdict

The IUT does not initiate at least one Data Length Update Procedure by sending an LL_LENGTH_REQ to the Lower Tester during this test case.



4.3.5.50 LL/CON/MAS/BV-76-C [Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY]

- Test Purpose

Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE 2M PHY.



- Test Procedure

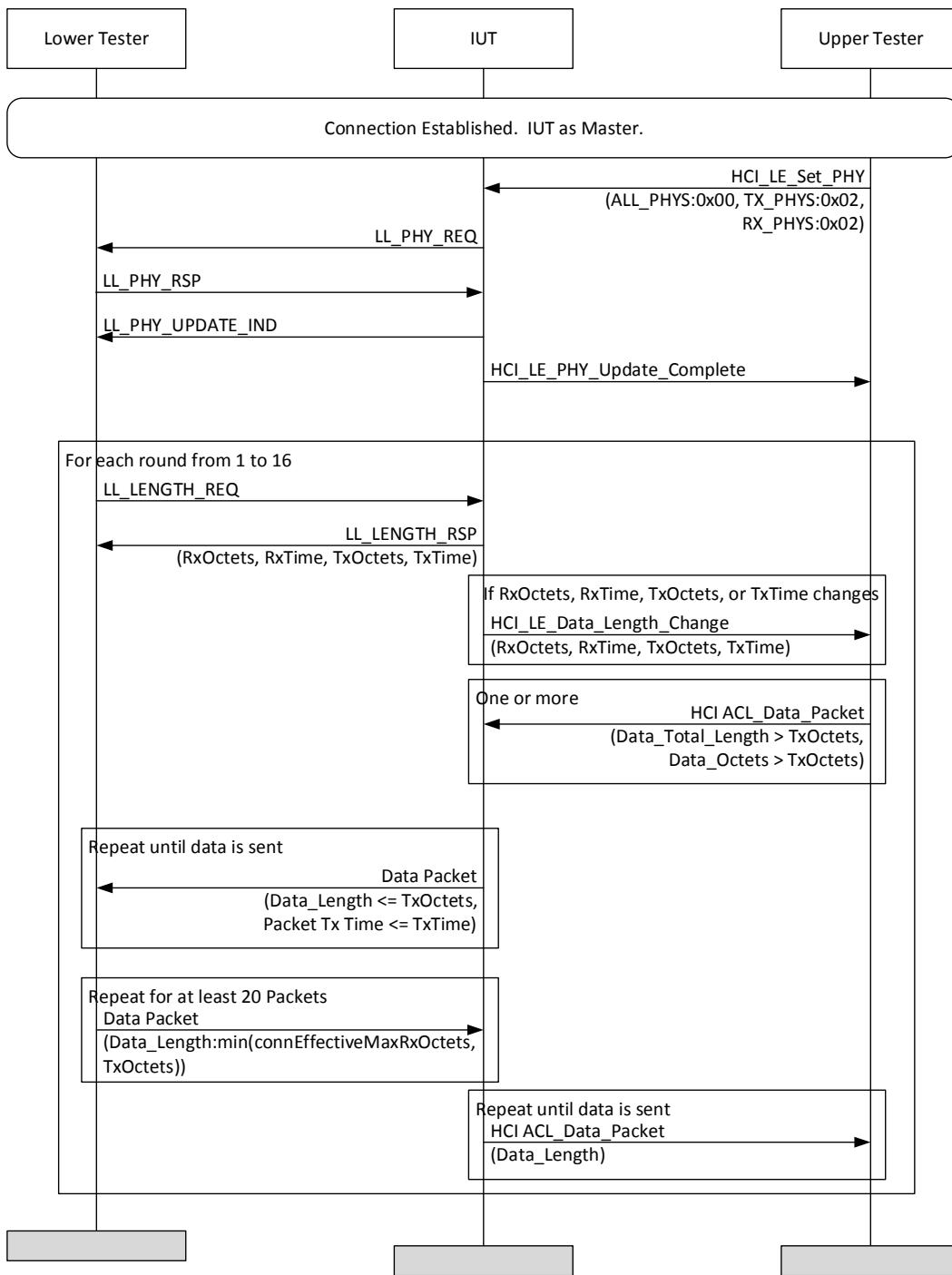


Figure 4.265: LL/CON/MAS/BV-76-C: Master Data Length Update – Responding to Data Length Update Procedure; LE 2M PHY

This test is identical to [LL/CON/MAS/BV-73-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY\]](#) except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x02 (LE 2M PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an



HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE 2M PHY, return an Inconclusive Verdict.

- Expected Outcome

The same as [LL/CON/MAS/BV-73-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY\]](#) and add the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE 2M PHY.

4.3.5.51 [LL/CON/MAS/BV-77-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY\]](#)

- Test Purpose

Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE 2M PHY.



- Test Procedure

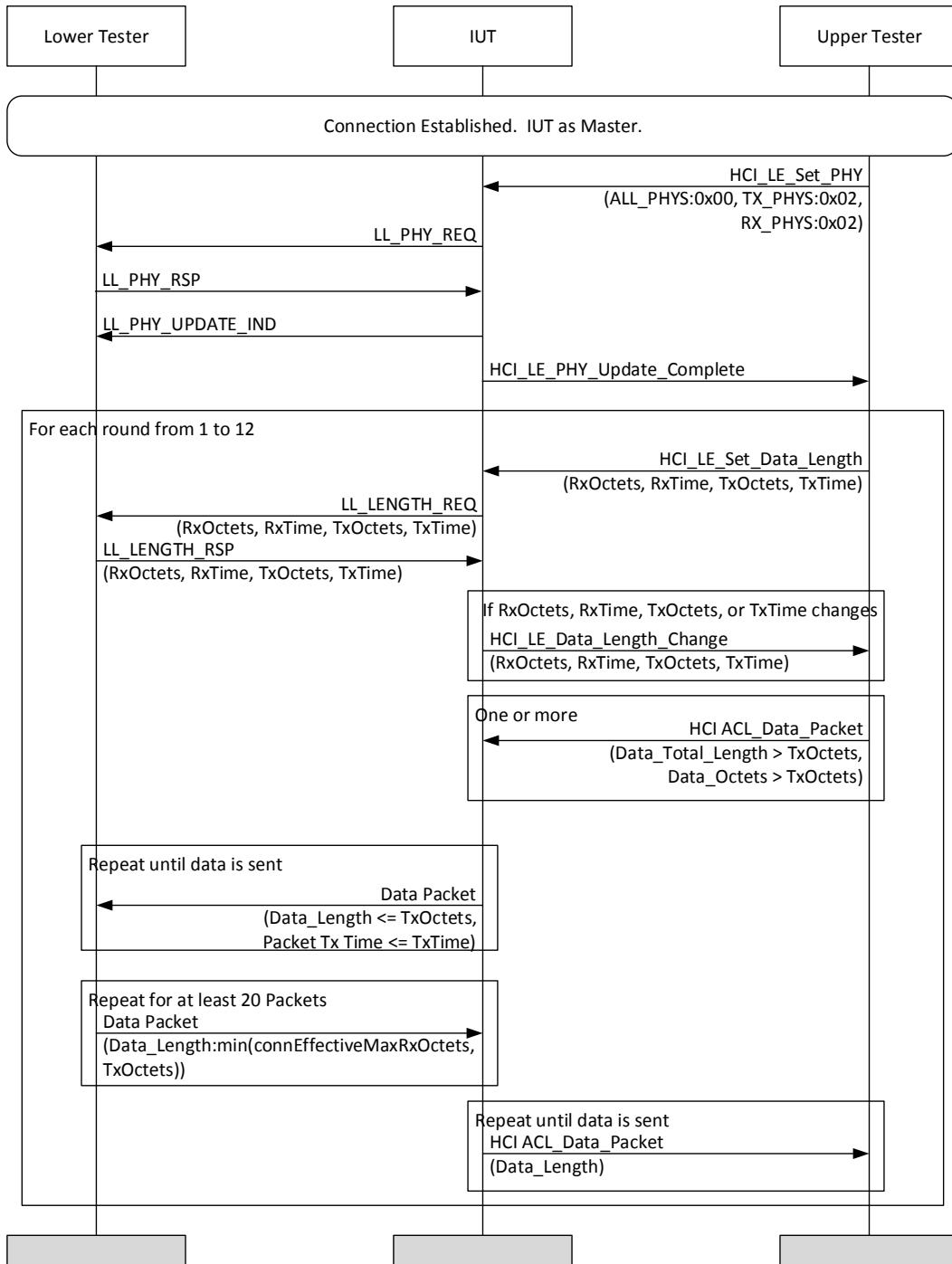


Figure 4.266: LL/CON/MAS/BV77-C: Master Data Length Update – Initiating Data Length Update Procedure; LE 2M PHY

This test is identical to [LL/CON/MAS/BV-74-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY\]](#) except that, before step 1, execute this following step:



- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x02 (LE 2M PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE 2M PHY, return an Inconclusive Verdict.
- Expected Outcome

The same as [LL/CON/MAS/BV-74-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY\]](#) and add in the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE 2M PHY.

4.3.5.52 [LL/CON/MAS/BV-78-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY\]](#)

- Test Purpose

Verify that the IUT as Master correctly handles reception of an LL_LENGTH_REQ PDU on the LE Coded PHY.



- Test Procedure

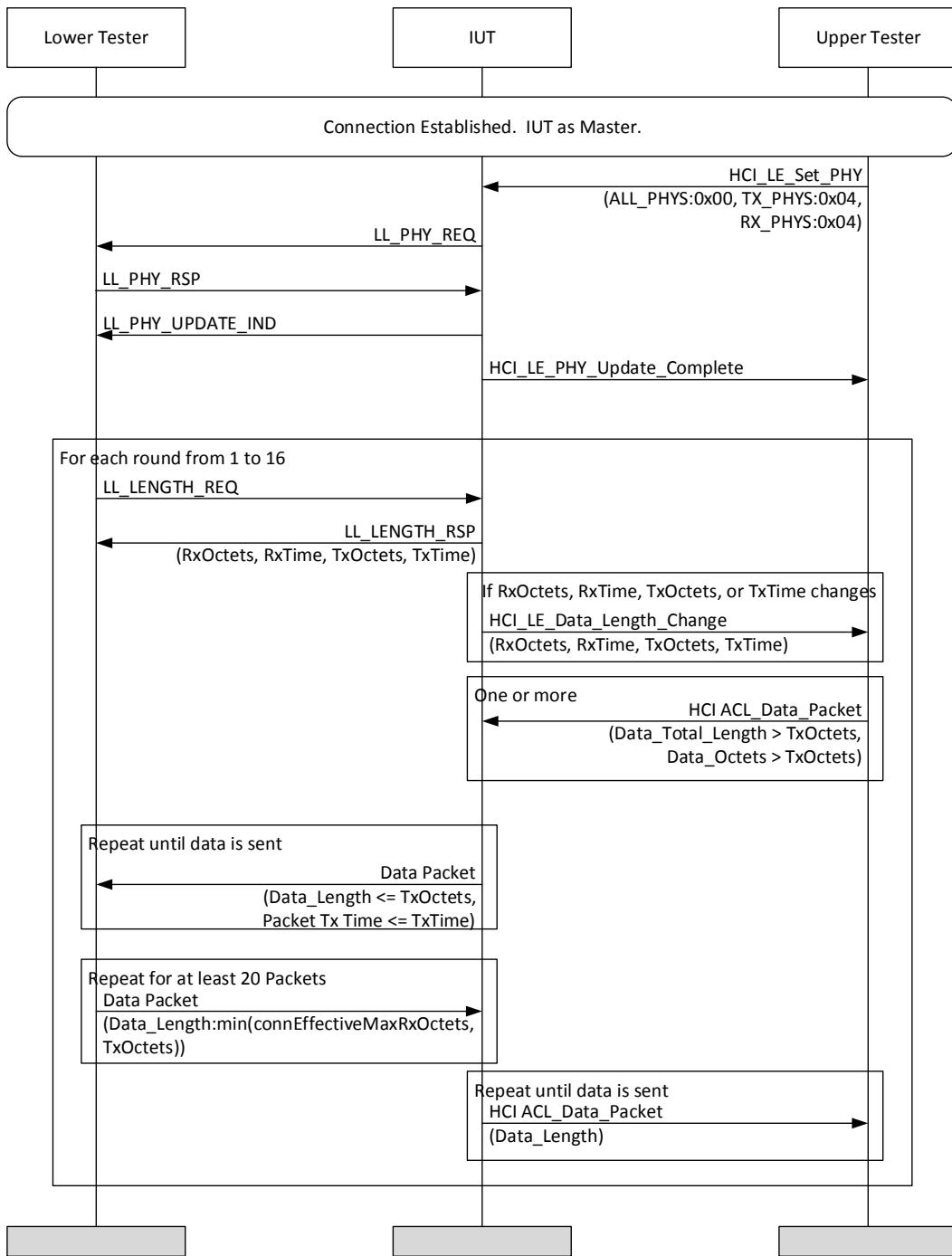


Figure 4.267: LL/CON/MAS/BV-78-C: Master Data Length Update – Responding to Data Length Update Procedure; LE Coded PHY

This test is identical to [LL/CON/MAS/BV-73-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY\]](#) except that, before step 1, execute this following step:

- The Upper Tester sends an **HCI LE Set PHY** command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x04 (LE Coded PHY only). The Lower Tester and IUT



then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.

- Expected Outcome

The same as [LL/CON/MAS/BV-73-C \[Master Data Length Update – Responding to Data Length Update Procedure; LE 1M PHY\]](#) and add in the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE Coded PHY.

4.3.5.53 [LL/CON/MAS/BV-79-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY\]](#)

- Test Purpose

Verify that a master IUT is able to perform the Data Length Update Procedure by sending an LL_LENGTH_REQ PDU on the LE Coded PHY.



- Test Procedure

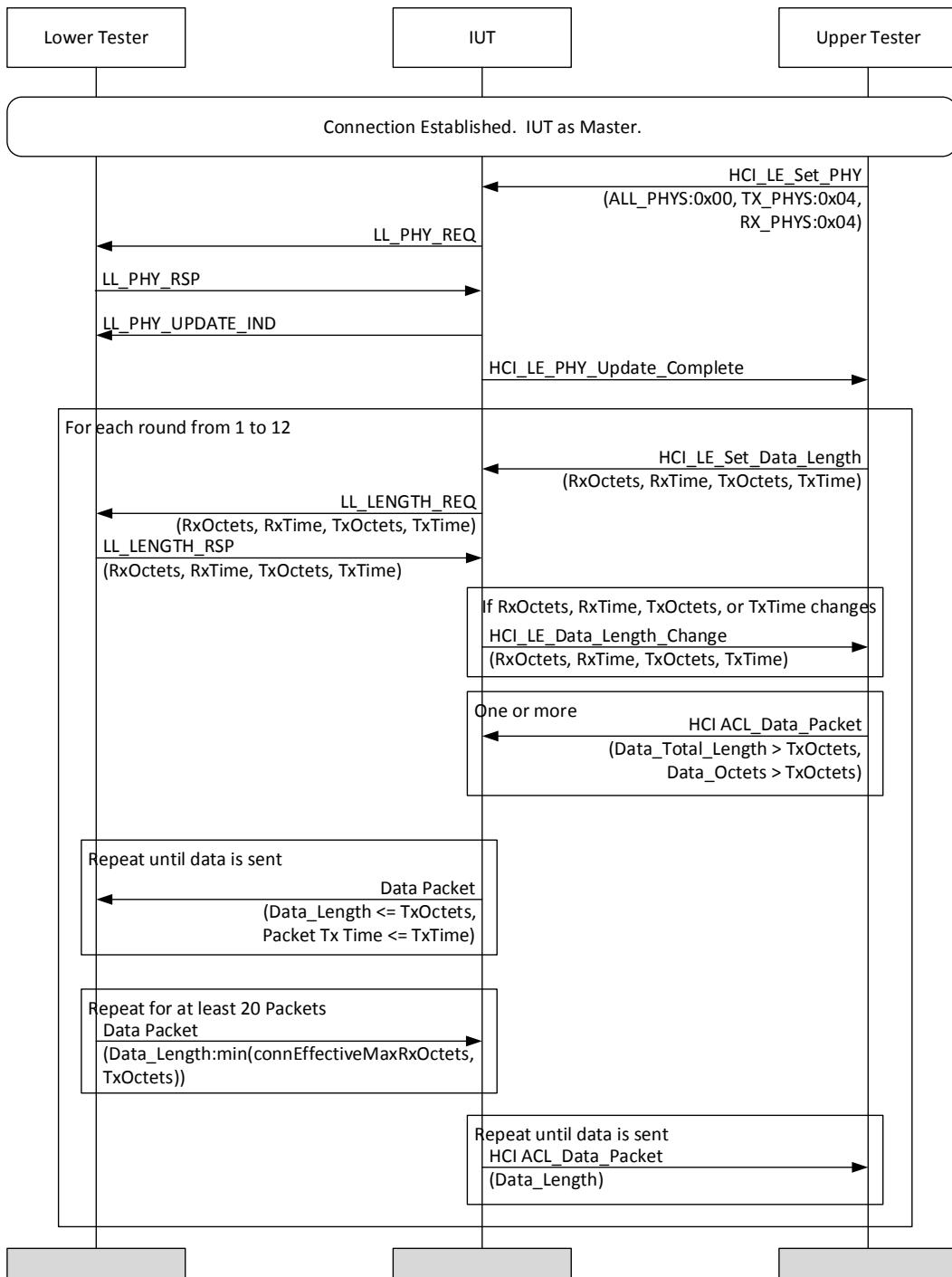


Figure 4.268: LL/CON/MAS/BV-79-C: Master Data Length Update – Initiating Data Length Update Procedure; LE Coded PHY

This test is identical to [LL/CON/MAS/BV-74-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY\]](#) except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x04 (LE Coded PHY only). The Lower Tester and IUT



then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.

- Expected Outcome

The same as [LL/CON/MAS/BV-74-C \[Master Data Length Update – Initiating Data Length Update Procedure; LE 1M PHY\]](#) and add in the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE Coded PHY.

4.3.5.54 [LL/CON/MAS/BV-80-C \[Master Data Length Update – Slave does not support; LE Coded PHY\]](#)

- Test Purpose

Verify that the IUT as Master correctly handles communication with a Lower Tester that does not support the Data Length Update Procedure on LE Coded PHY.



- Test Procedure

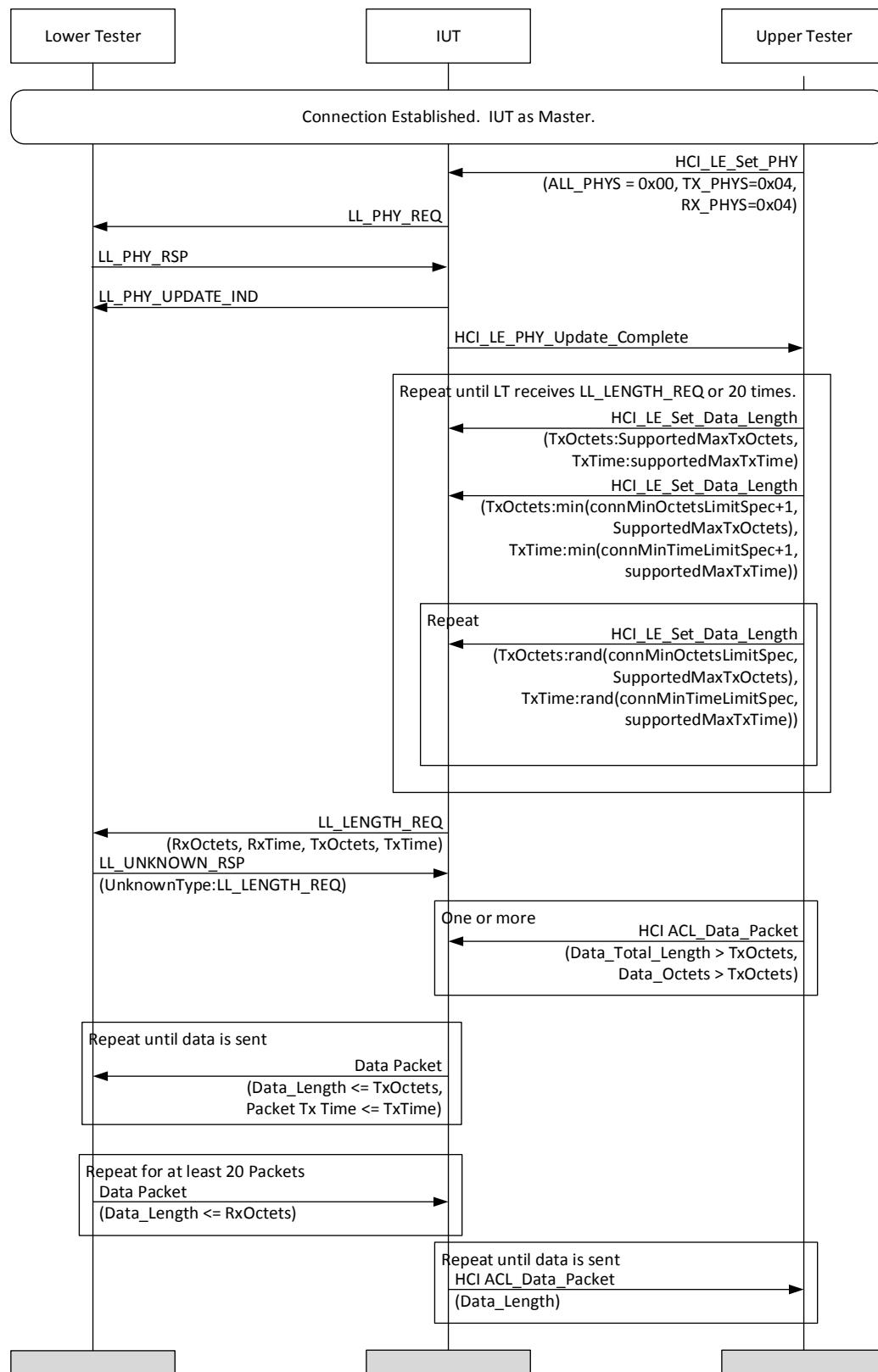


Figure 4.269: LL/CON/MAS/BV-80-C: Master Data Length Update – Slave does not support; LE Coded PHY



This test is identical to [LL/CON/MAS/BV-75-C \[Master Data Length Update – Slave does not support; LE 1M PHY\]](#) except that, before step 1, execute this following step:

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to zero and both TX_PHYS and RX_PHYS set to 0x04 (LE Coded PHY only). The Lower Tester and IUT then complete the PHY Update Procedure. The Upper Tester shall expect the IUT to send an HCI_LE_PHY_Update_Complete event. If the PHY in both directions is not changed to or left as the LE Coded PHY, return an Inconclusive Verdict.
- Expected Outcome

The same as [LL/CON/MAS/BV-75-C \[Master Data Length Update – Slave does not support; LE 1M PHY\]](#) and add in the following Inconclusive Verdict:

Inconclusive Verdict

The IUT does not change to the LE Coded PHY.

4.3.5.55 LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when a feature exchange has not been performed and the remote device does not support the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and sends an LL_UNKNOWN_RSP PDU in response to the IUT's request.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

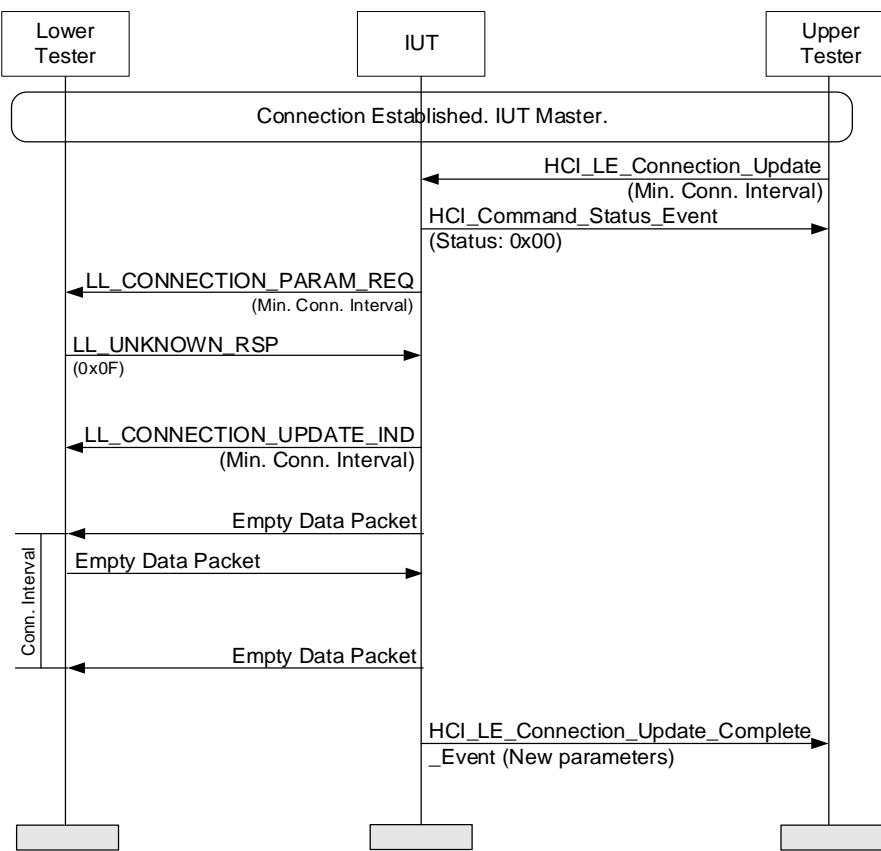


Figure 4.270: LL/CON/MAS/BV-81-C [Initiating Connection Parameter Request – Unsupported Without Feature Exchange]

- If the IUT autonomously initiates a feature exchange before step 3, the test ends with an Inconclusive Verdict
- Upper Tester sends an HCI LE Connection Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI Command Status event from the IUT in response.
- Lower Tester expects an LL CONNECTION PARAM REQ control PDU from the IUT (the actual parameters in the LL CONNECTION PARAM REQ PDU may be different from the parameters provided by the Upper Tester). Lower Tester responds with an LL UNKNOWN RSP PDU with the Opcode field set to LL CONNECTION PARAM REQ (0x0F).
- Lower Tester expects an LL CONNECTION UPDATE IND PDU from the IUT (the actual parameters in the LL CONNECTION UPDATE IND PDU may be different from the parameters provided by the Upper Tester).
- Maintain the connection using empty data packets until the event count matches the time indicated in the connection update request packet.
- Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
- Upper Tester expects an HCI LE Connection Update Complete from the IUT containing the new connection parameters.



8. Repeat steps 2 and 4-7, using connection parameter values in step 2 different than in the first iteration. If the IUT sends an LL_CONNECTION_PARAM_REQ PDU in this step, the test ends with a Fail Verdict.
- Expected Outcome

Pass Verdict

The IUT transmits the LL_CONNECTION_PARAM_REQ PDU to update connection parameters, then it sends an LL_CONNECTION_UPDATE_IND PDU.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.

In the second iteration, the IUT skips the LL_CONNECTION_PARAM_REQ PDU and sends an LL_CONNECTION_UPDATE_IND PDU.

4.3.5.56 LL/CON/MAS/BV-82-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure after the feature exchange reveals that the remote device does not support the request.

The Lower Tester acts in the slave role maintaining a connection and initiates feature exchange, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT.

- Reference

[8] 5.1.7

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval greater than LL_slave_connSlaveLatency_MIN up to LL_slave_connSlaveLatency_MAX, common slave latency, common timeout).



- Test Procedure

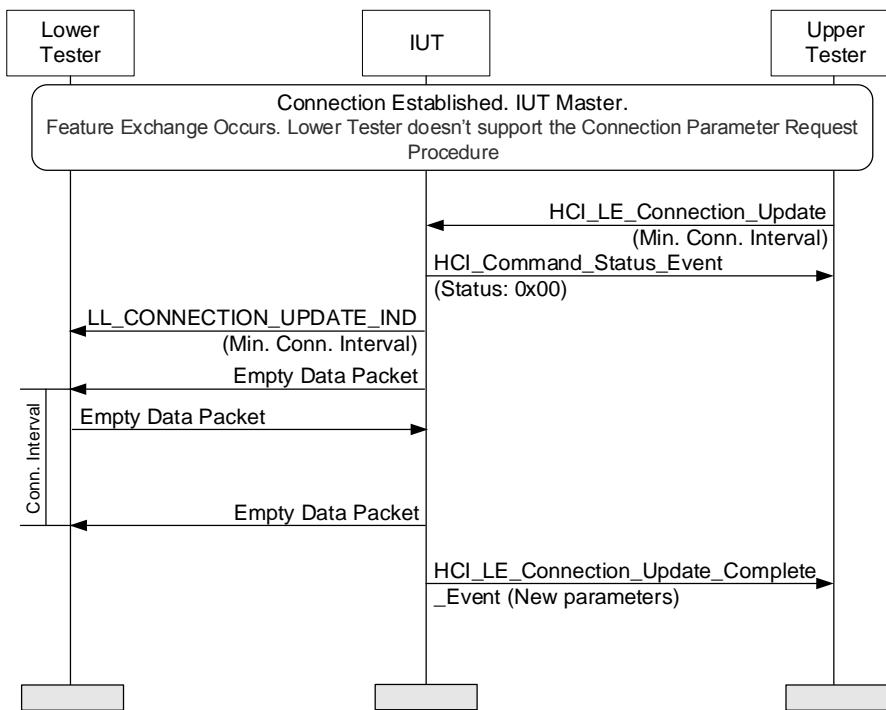


Figure 4.271: LL/CON/MAS/BV-82-C [Initiating Connection Parameter Request – Unsupported With Feature Exchange]

1. Lower Tester initiates a feature exchange, unless the IUT has already done so. Lower Tester indicates that it does not support the Connection Parameter Request Procedure.
2. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
3. Lower Tester expects an LL_CONNECTION_UPDATE_IND PDU from the IUT (the actual parameters in the LL_CONNECTION_UPDATE_IND PDU may be different from the parameters provided by the Upper Tester). If the IUT sends an LL_CONNECTION_PARAM_REQ PDU instead of the LL_CONNECTION_UPDATE_IND PDU, the test ends with a Fail Verdict.
4. Maintain the connection using empty DATA packets until the event count matches the time indicated in the connection update request packet.
5. Once the event count matches the time, the new parameters such as the new connection interval selected by the IUT will be used.
6. Upper Tester expects an HCI_LE_Connection_Update_Complete from the IUT containing the new connection parameters.

- Expected Outcome

Pass Verdict

The IUT transmits an LL_CONNECTION_UPDATE_IND PDU.

The IUT reports the selected new connection parameters with an HCI_LE_Connection_Update_Complete event.



4.3.5.57 LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]

- Test Purpose

Test that a master IUT does not change the PHY when it requests an asymmetrical PHY connection, but the slave indicates it only supports a symmetrical connection.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the PHY update procedure as the Host of the IUT, and the Lower Tester then observes the PHY update procedure carried out by the IUT.

- Reference

[10] 5.1.10

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout). All communications shall use the 1M PHY.

- Test Procedure

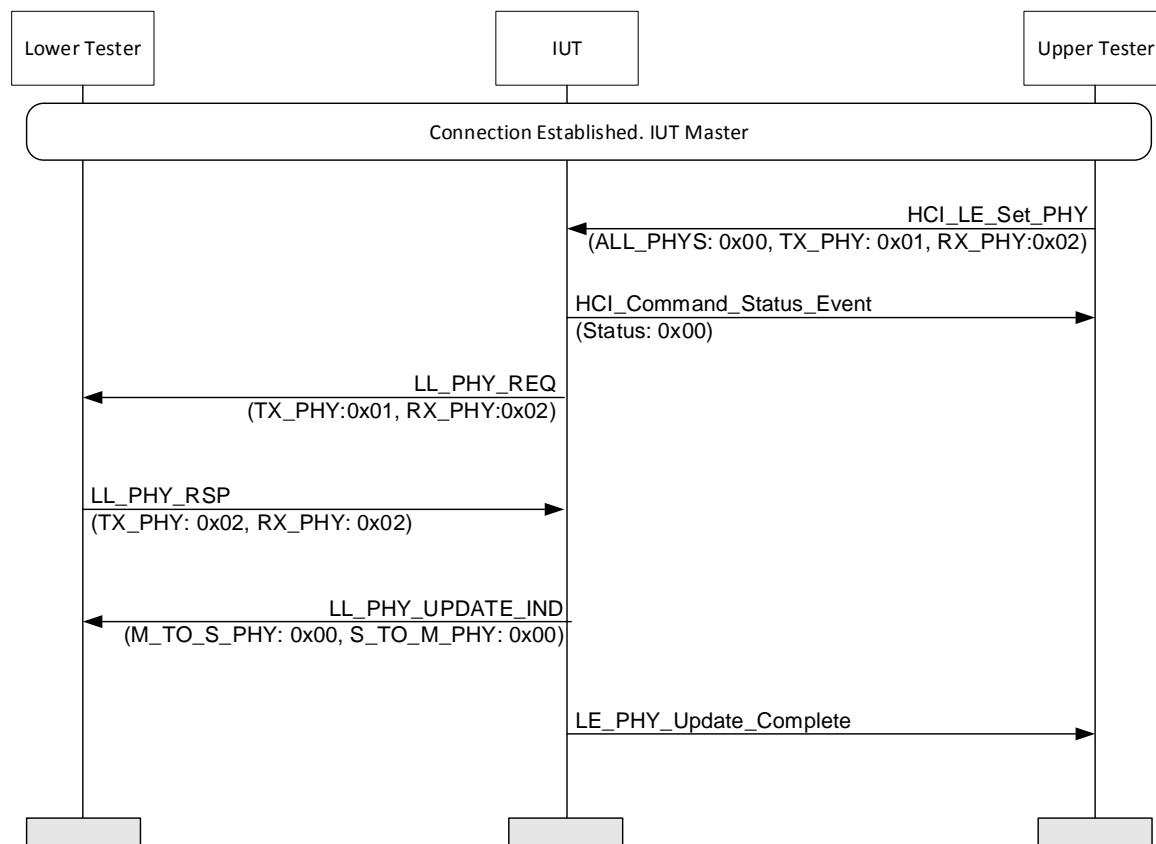


Figure 4.272: LL/CON/MAS/BV-117-C [PHY Update Procedure – Master Requests Asymmetrical, Slave Symmetrical]

- The Upper Tester sends an HCI_LE_Set_PHY command to the IUT with ALL_PHYS set to 0x00, TX_PHY set to 0x01, and RX_PHY set to 0x02.
- The Upper Tester expects an HCI_Command_Status event from the IUT in response.



3. The IUT initiates a PHY change by issuing an LL_PHY_REQ command to the Lower Tester with TX_PHY set to 0x01 and RX_PHY set to 0x02. If the IUT does not send an LL_PHY_REQ PDU or sends LL_PHY_REQ PDU with different values in TX_PHY or RX_PHY, the test case ends with an Inconclusive Verdict.
 4. The Lower Tester responds with LL_PHY_RSP with TX_PHY set to 0x02 and TX_PHY set to 0x02.
 5. The IUT sends an LL_PHY_UPDATE_IND to the Lower Tester with M_TO_S_PHY set to 0x00 and S_TO_M_PHY set to 0x00.
 6. The Upper Tester expects a LE_PHY_Update_Complete event from the IUT with a payload indicating no change.
- Expected Outcome

Pass Verdict

The IUT completes the sequence by issuing an LL_PHY_UPDATE_IND with M_TO_S_PHY set to 0x00 and S_TO_M_PHY set to 0x00, indicating no change.

Inconclusive Verdict

The IUT does not initiate a PHY Update Procedure by sending an LL_PHY_REQ PDU or sends sending an LL_PHY_REQ PDU with TX_PHY not set to 0x01 and RX_PHY not set to 0x02 in step 3).

4.3.5.58 LL/CON/MAS/BI-02-C [Master T_Terminate Timer]

- Test Purpose

Test the correct behavior of a master device when TERMINATE_IND packets are not acknowledged.

The Lower Tester acts in the slave role, expects TERMINATE_IND packets from the IUT and does not acknowledge them, until T_Terminate expires.

- Reference

[3] 5.1.6

- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using the common connection parameters.



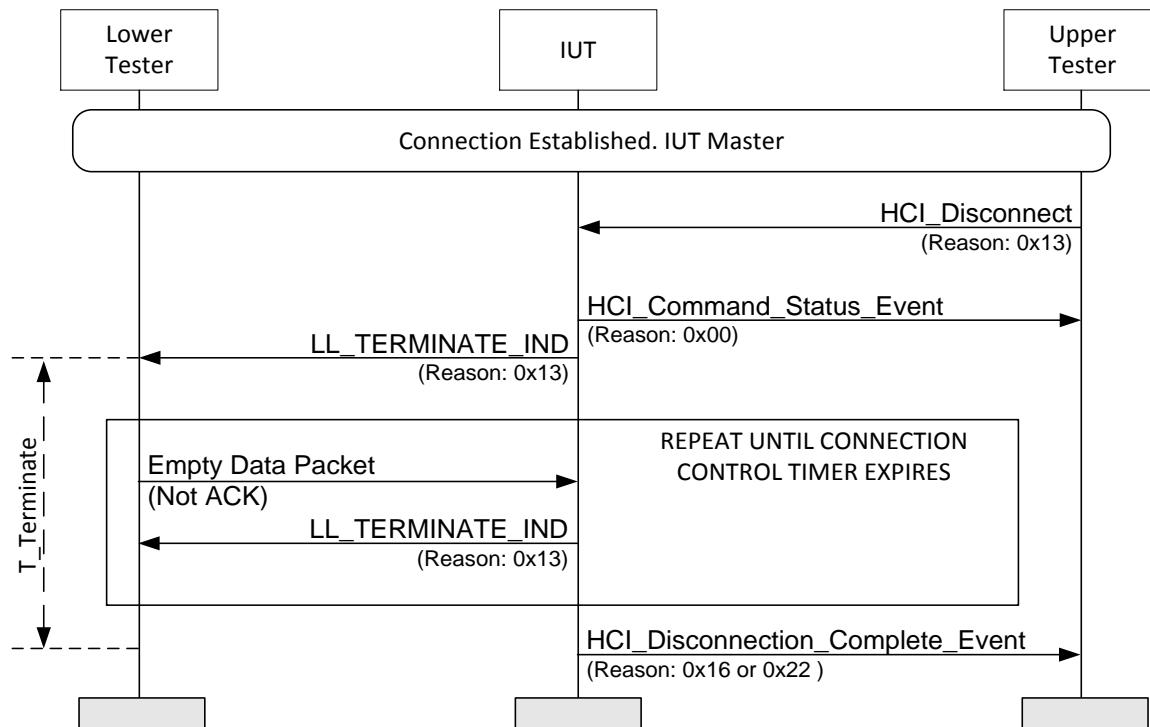


Figure 4.273: LL/CON/MAS/BI-02-C [Master T_Terminate Timer]

- Upper Tester sends an HCI_Disconnect command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status in response.
 - Lower Tester expects the IUT to transmit a TERMINATE_IND packet. Lower Tester does not acknowledge the termination packet, but continues the slave transmissions.
 - Lower Tester expects the retransmission of TERMINATE_IND packet from the IUT and responds with an empty DATA packet not acknowledging the termination packet. Repeat until T_Terminate expires.
 - Upper Tester expects an HCI_Disconnection_Complete event including status of 0x00 (success), and a reason code of 0x16 ("Connection Terminated by Local Host") or 0x22 ("LL Response Timeout") from the IUT indicating loss of the link.
- Expected Outcome

Pass Verdict

The IUT reports the connection termination with an HCI event.

The IUT keeps sending TERMINATE_IND packets until T_Terminate timer expires.

4.3.5.59 LL/CON/MAS/BI-04-C [Master Connection Control Timer]

- Test Purpose

Test that a slave device is able to recover from a control procedure failure.

The Lower Tester acts in the slave role, failing to acknowledge a control procedure request in order to expire the master connection control timer.

- Reference

[3] 5.2



- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using the common connection parameters.

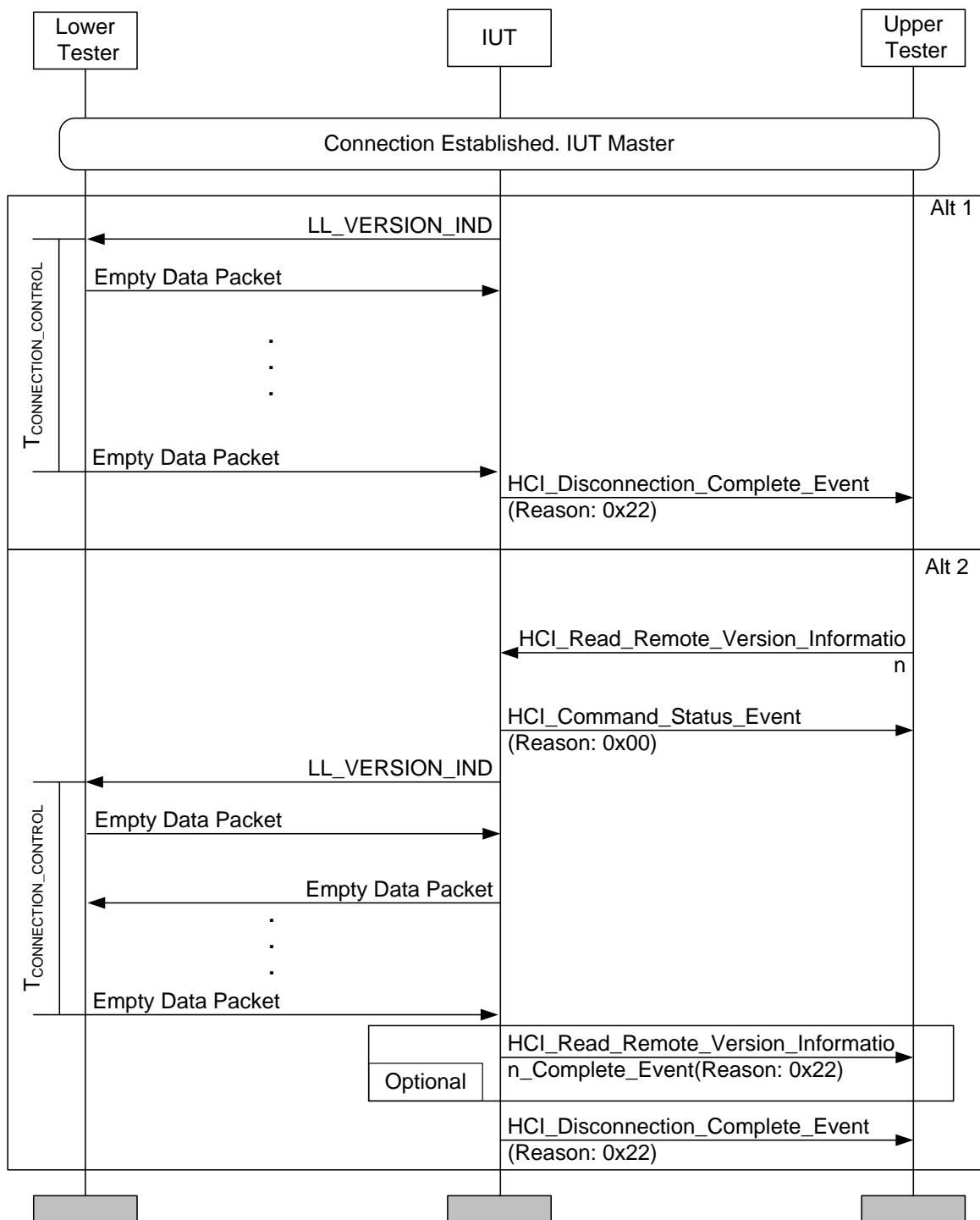


Figure 4.274: LL/CON/MAS/BI-04-C [Master Connection Control Timer]



1. Once connection is established, if the IUT has sent LL_VERSION_IND packet by itself, Lower Tester acknowledges the version packet. Lower Tester continues the slave transmissions but it never sends the LL_VERSION_IND packet. Then, continue on Step 4.
 2. If the IUT does not send LL_VERSION_IND packet by itself, Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT containing the connection handle from the preamble steps' execution and expects an HCI_Command_Status in response.
 3. Lower Tester expects the IUT to transmit a VERSION_IND packet. Lower Tester acknowledges the version packet.
 4. Lower Tester continues sending empty DATA packet until the connection control timeout value, or until Step 5 executes.
 5. Upper Tester expects an HCI_Disconnection_Complete event from the IUT indicating loss of the link with connection handle matching that of the preamble steps and the error code indicating termination from LL response timeout.
- Expected Outcome

Pass Verdict

The test procedure executes successfully with the IUT using the connection control timer to recover from the failed control procedure.

4.3.5.60 LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]

- Test Purpose

Test that a master IUT is able to perform the connection parameter request procedure when the remote device does not respond to the request.

The Lower Tester acts in the slave role maintaining a connection, the Upper Tester issues the HCI command to start the connection parameter request procedure as the Host of the IUT, and the Lower Tester then observes the procedure carried out by the IUT and does not respond to the IUT's request.

- Reference
- [3] 5.1.7
- Initial Conditions

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX,
 LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN,
 LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

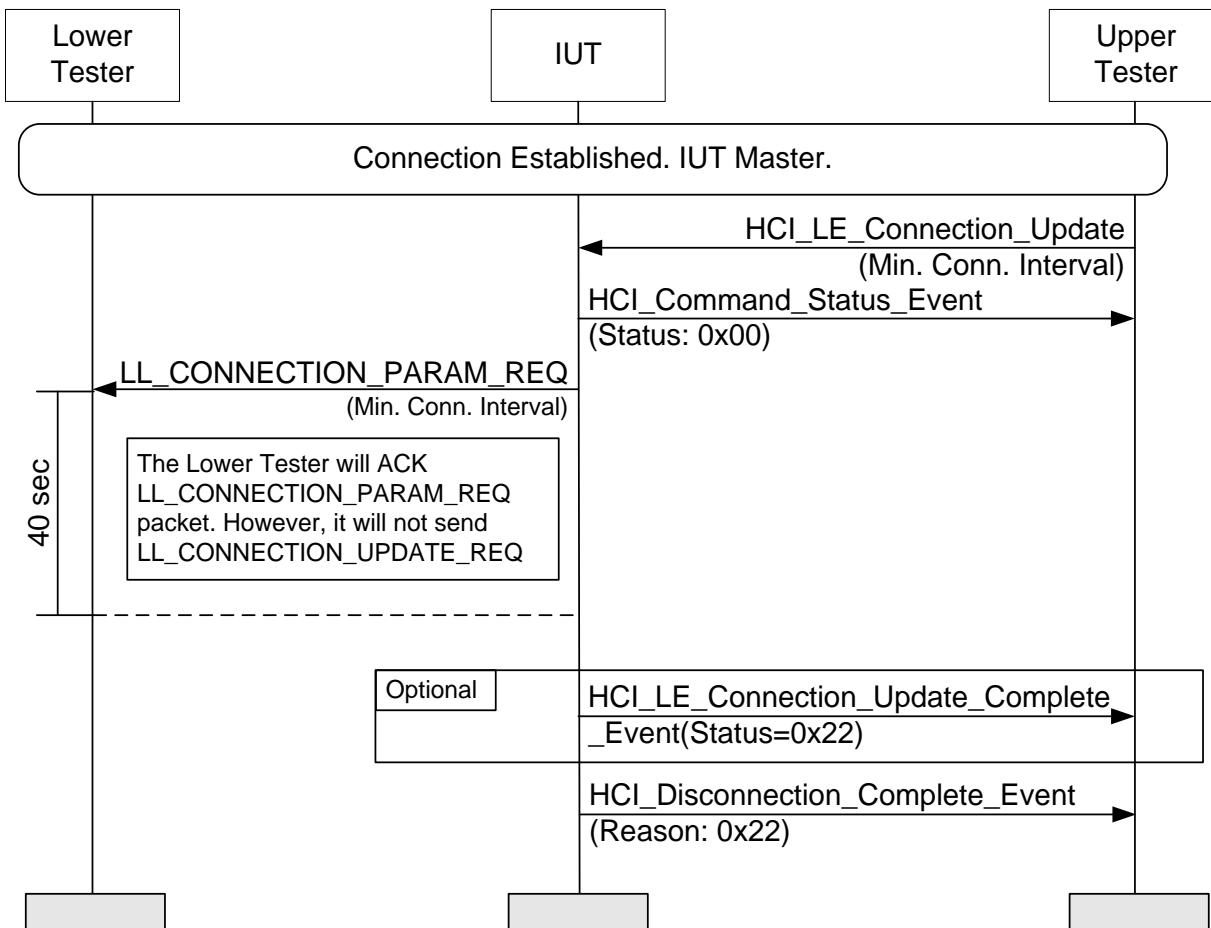


Figure 4.275: LL/CON/MAS/BI-05-C [Initiating Connection Parameter Request – Timeout]

1. Upper Tester sends an HCI_LE_Connection_Update command to the IUT setting the connection parameters to the minimum connection interval, no latency and intermediate connection supervision timeout (3 s) and expects an HCI_Command_Status event from the IUT in response.
2. Lower Tester expects an LL_CONNECTION_PARAM_REQ control PDU from the IUT (the actual parameters in the LL_CONNECTION_PARAM_REQ PDU may be different from the parameters provided by the Upper Tester). The Lower Tester acknowledges the LL_CONNECTION_PARAM_REQ PDU but does not respond to the LL_CONNECTION_PARAM_REQ.
3. IUT sends the HCI_Disconnect_Complete event with reason code set to 0x22 (LL Response Timeout) to the Upper Tester and the IUT stops maintaining the connection.

- Expected Outcome

Pass Verdict

The IUT sends HCI_Disconnection_Complete_Event (Reason: 0x22) when connection control transaction timer expires and the IUT stops maintaining the connection.

4.3.5.61 LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]

- Test Purpose

Test that a master IUT is able to respond to a connection parameter request procedure from a slave device when the connection parameter request from the slave contains illegal parameters.



- Reference

[3] 5.1.7

- Initial Conditions

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

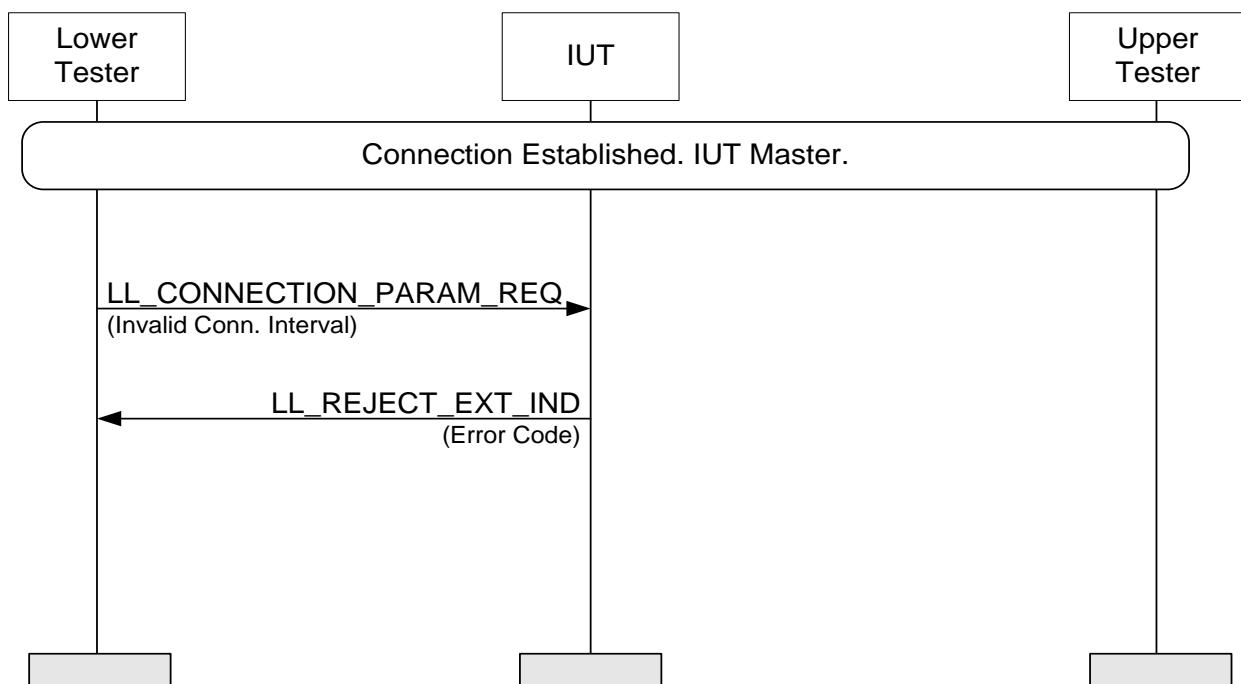


Figure 4.276: LL/CON/MAS/BI-06-C [Accepting Connection Parameter Request – illegal parameters]

1. Lower Tester sends an LL_CONNECTION_PARAM_REQ PDU to the IUT setting the connection interval min and max to outside the valid range i.e. 4 (6ms).

2. Lower Tester expects an LL_REJECT_EXT_IND control PDU from the IUT with ErrorCode 0x1E.

- Expected Outcome

Pass Verdict

The IUT responds to the Lower Tester's request to update connection parameters with an LL_REJECT_EXT_IND using the correct ErrorCode.



4.3.6 Both Connected Roles

4.3.6.1 Constant Tone Extension Request Procedure, IUT Responding, Unsupported

- Test Purpose

Tests that the IUT responds correctly to a Constant Tone Extension Request Procedure initiated by the Lower Tester when the IUT does not support the proper CTE type or length. Test that the IUT generates an LL_REJECT_EXT_IND PDU.

The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT for each parameter and confirms the IUT correctly sends an LL_REJECT_EXT_IND PDU.

- Test Case IDs

[LL/CON/SLA/BV-87-C](#)

[LL/CON/MAS/BV-83-C](#)

- Reference

[\[10\] 5.1.12](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- Test Procedure

Execute the test procedure for IUT as Master (if supported) and IUT as Slave (if supported).



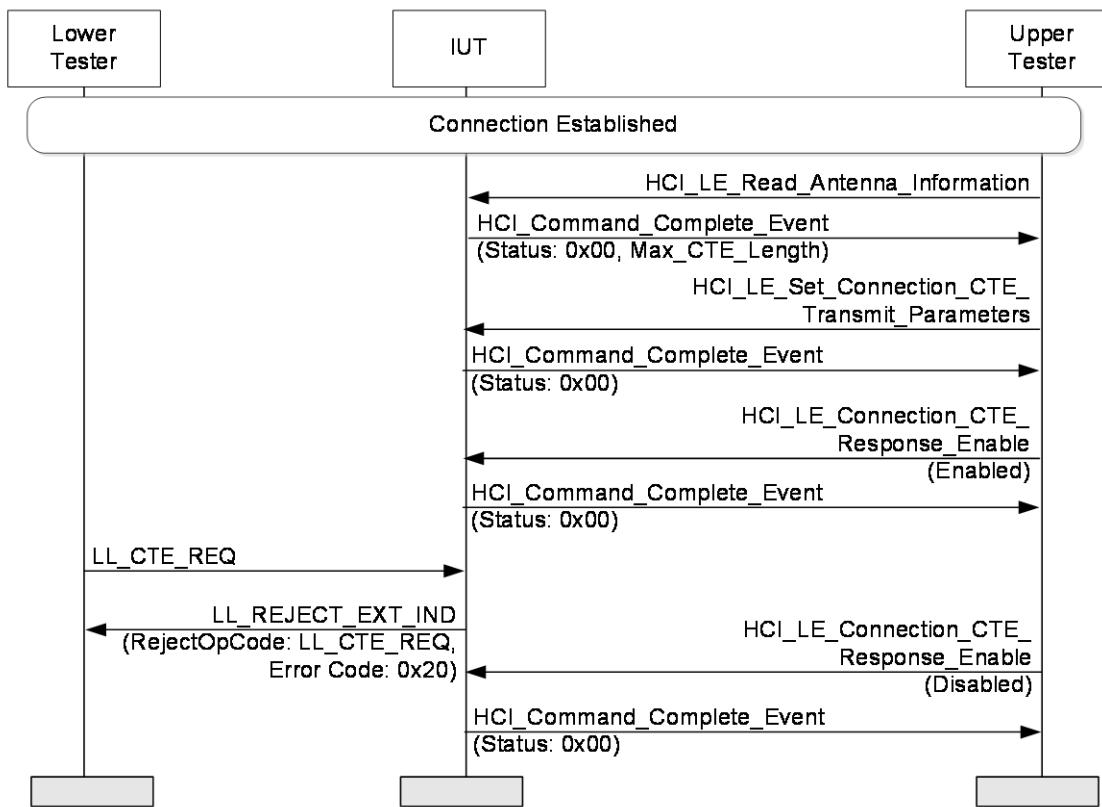


Figure 4.277: [Constant Tone Extension Request Procedure, IUT Responding, Unsupported]

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. If the IUT supports AoD, the Upper Tester expects the IUT to also return a Max_Length_Switching_Pattern between 0x02 and 0x4B. The Upper Tester stores the Max_CTE_Length and the Max_Length_Switching_Pattern (if applicable) for future use.
2. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Transmit_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. CTE_Types shall be set to a supported value. If AoD is selected, Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values.
3. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).
4. If the Max_CTE_Length value is equal to 0x14, skip to step 7.
5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to Max_CTE_Length + 1. CTETypeReq is set to a value that matches the CTE_Types value specified in step 2.
6. The Lower Tester expects an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).
7. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to 0x02. CTETypeReq is set to a value that does not match the CTE_Types value specified in step 2.



8. The Lower Tester expects an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).
 9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).
- Expected Outcome

Pass Verdict

The IUT sends an LL_REJECT_EXT_IND PDU with RejectOpCode set to LL_CTE_REQ and ErrorCode set to 0x20 (Unsupported LMP Parameter Value/ Unsupported LL Parameter Value).

4.3.6.2 [Periodic Advertising Sync Transfer Procedure, Advertising IUT Initiated]

- Test Purpose

Tests that the IUT with periodic advertising enabled is able to initiate the Periodic Advertising Sync Transfer Procedure. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well. The test purpose is exercised over each supported PHY, including using different PHYs for the connection and periodic advertising. The case that the remote does not support the procedure is tested.

The Lower Tester verifies that the IUT continues periodic advertising and that the IUT transfers synchronization information that is valid for that periodic advertising. The Upper Tester commands the IUT to perform the procedure and observes the responses returned by the IUT.

- Reference

[13] 4.6.23

- Initial Condition

MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.45](#) for each applicable Test Case ID.

Test Case ID	Connection PHY	Primary ADV PHY	Secondary ADV PHY	Test Privacy and Unknown Rsp
	Step 1	Step 2	Step 2, Step 10	Step 15
LL/CON/SLA/BV-88-C LL/CON/MAS/BV-84-C	LE 1M	0x01 (LE 1M)	0x01 (LE 1M)	Yes
LL/CON/SLA/BV-89-C LL/CON/MAS/BV-85-C	LE 1M	0x03 (LE Coded)	0x03 (LE Coded)	No
LL/CON/SLA/BV-90-C LL/CON/MAS/BV-86-C	LE 1M	0x01 (LE 1M)	0x02 (LE 2M)	No
LL/CON/SLA/BV-91-C LL/CON/MAS/BV-87-C	LE Coded	0x01 (LE 1M)	0x01 (LE 1M)	No



Test Case ID	Connection PHY	Primary ADV PHY	Secondary ADV PHY	Test Privacy and Unknown Rsp
	Step 1	Step 2	Step 2, Step 10	Step 15
LL/CON/SLA/BV-92-C LL/CON/MAS/BV-88-C	LE 2M	0x01 (LE 1M)	0x01 (LE 1M)	No

Table 4.45: Values for each Test Case ID



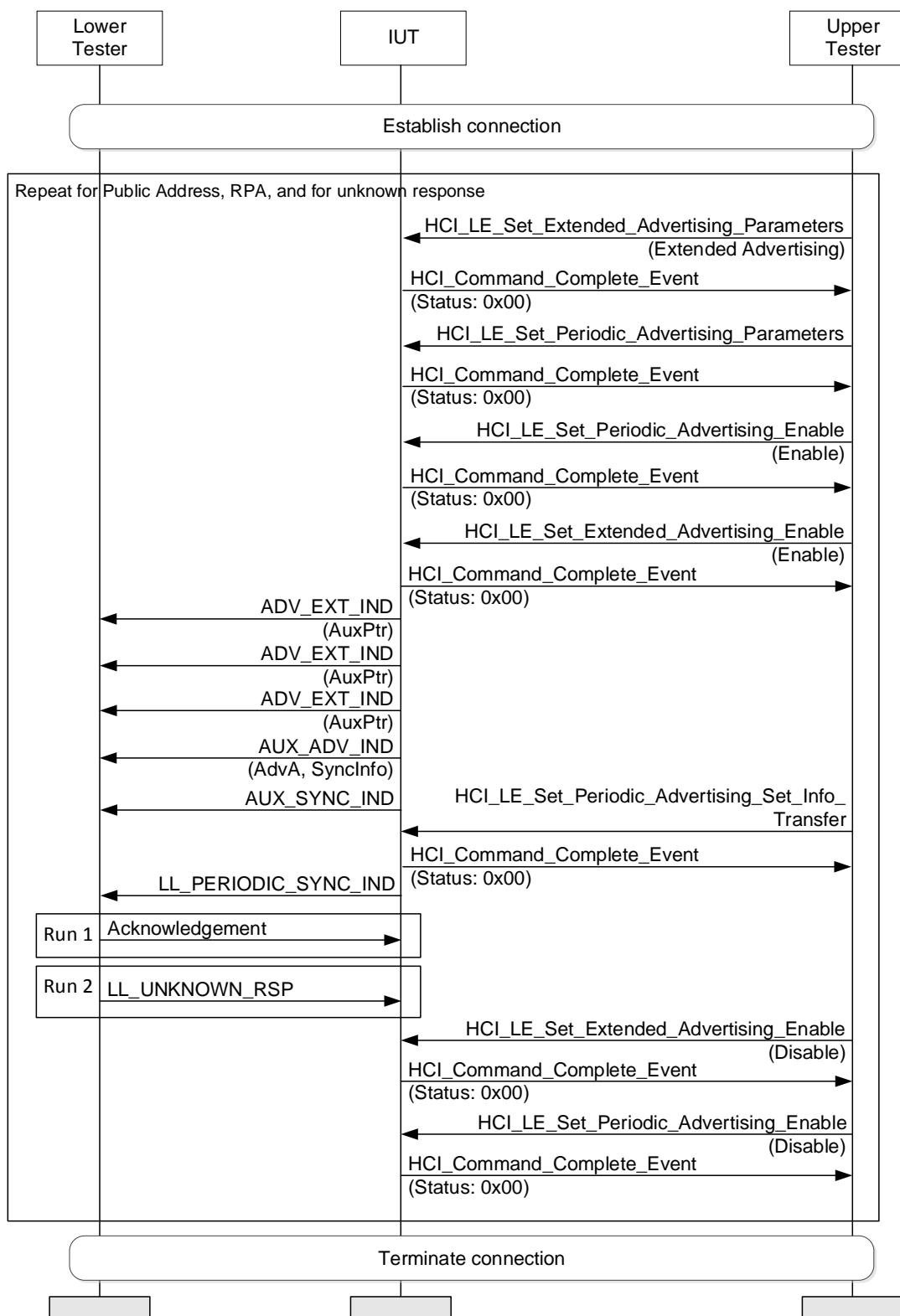


Figure 4.278: [Periodic Advertising Sync Transfer Procedure, Advertising IUT Initiated]



1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.45](#), and the Lower Tester connects with the IUT on that PHY.
2. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Parameters command to the IUT. Advertising_Event_Properties parameter is set to 0x0000. Own_Address_Type is set to 0x00 (Public Address). The Primary_Advertising_PHY is set to the value indicated in the Primary ADV PHY column in [Table 4.45](#). The Secondary_Advertising_PHY is set to the value indicated in the Secondary ADV PHY column in [Table 4.45](#).
3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Parameters command to the IUT. Periodic_Advertising_Properties parameter is set to 0x0000.
4. The Upper Tester enables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
5. The Upper Tester enables advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
6. The Lower Tester expects an ADV_EXT_IND PDU from the IUT with the AuxPtr Extended Header field present.
7. The Lower Tester utilizes the AuxPtr field to listen for an AUX_ADV_IND PDU on the secondary advertising channel with the SyncInfo Extended Header fields present.
8. The Lower Tester utilizes the SyncInfo field to listen for an AUX_SYNC_IND PDU on the secondary advertising channel and synchronizes with the periodic advertisements.
9. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Set_Info_Transfer command to the IUT with the Advertising_Handle of the advertising configured in step 2, Service_Data set to any non-zero value, and the Connection_Handle of the connection and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
10. The Lower Tester expects an LL_PERIODIC_SYNC_IND PDU from the IUT with ID set to the value used for Service_Data in step 9, SyncInfo and connEventCount set to valid values, SID set to match the advertising set ID representing the advertising in step 2, ATType set to the address type used for the advertising in step 2, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in [Table 4.45](#), and the AdvA set to the IUT's advertiser address used for the advertising in step 2.
11. The Lower Tester verifies that the IUT continues maintaining the connection.
12. The Lower Tester uses the SyncInfo and connEventCount values to verify that they allow synchronization to the periodic advertising as in step 8.
13. The Upper Tester disables extended advertising using the HCI_LE_Set_Extended_Advertising_Enable command.
14. The Upper Tester disables periodic advertising using the HCI_LE_Set_Periodic_Advertising_Enable command.
15. If the Test Privacy and Unknown Rsp column in [Table 4.45](#) indicates No, then skip to step 18.
16. If the IUT supports controller-based privacy, then repeat steps 2-14, except that in step 2 the Own_Address_Type is set to 0x02 (Resolvable Private Address) and Peer_Address_Type and Peer_Address are set to values matching the Lower Tester's address matching a local IRK in the resolving list, and that in step 10, the AdvA is set to the IUT's unresolved advertising address used for the advertising in step 2.
17. Repeat steps 2-11 and 13-14, except that in step 10 the Lower Tester sends an LL_UNKNOWN_RSP PDU to the IUT with UnknownType set to LL_PERIODIC_SYNC_IND PDU and verifies that the IUT continues maintaining the connection.
18. Terminate the connection between the IUT and the Lower Tester.



- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.

4.3.6.3 [Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]

- Test Purpose

Tests that the IUT with periodic advertising enabled is able to initiate the Periodic Advertising Sync Transfer Procedure. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well. The test purpose is exercised over each supported PHY, including using different PHYs for the connection and periodic advertising. The case that the remote does not support the procedure is tested.

The Lower Tester verifies that the IUT continues periodic advertising and that the IUT transfers synchronization information that is valid for that periodic advertising. The Upper Tester commands the IUT to perform the procedure and observes the responses returned by the IUT.

- Reference

[13] 4.6.23

- Initial Condition

MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.46](#) for each applicable Test Case ID.

Test Case ID	Connection PHY	Primary ADV PHY	Secondary ADV PHY	Test Privacy
	Step 1	Step 2, Step 4	Step 2, Step 11	Step 17
LL/CON/SLA/BV-93-C LL/CON/MAS/BV-89-C	LE 1M	0x01 (LE 1M)	0x01 (LE 1M)	Yes
LL/CON/SLA/BV-94-C LL/CON/MAS/BV-90-C	LE 1M	0x03 (LE Coded)	0x03 (LE Coded)	No
LL/CON/SLA/BV-95-C LL/CON/MAS/BV-91-C	LE 1M	0x01 (LE 1M)	0x02 (LE 2M)	No
LL/CON/SLA/BV-96-C LL/CON/MAS/BV-92-C	LE Coded	0x01 (LE 1M)	0x01 (LE 1M)	No
LL/CON/SLA/BV-97-C LL/CON/MAS/BV-93-C	LE 2M	0x01 (LE 1M)	0x01 (LE 1M)	No

Table 4.46: Values for each Test Case ID



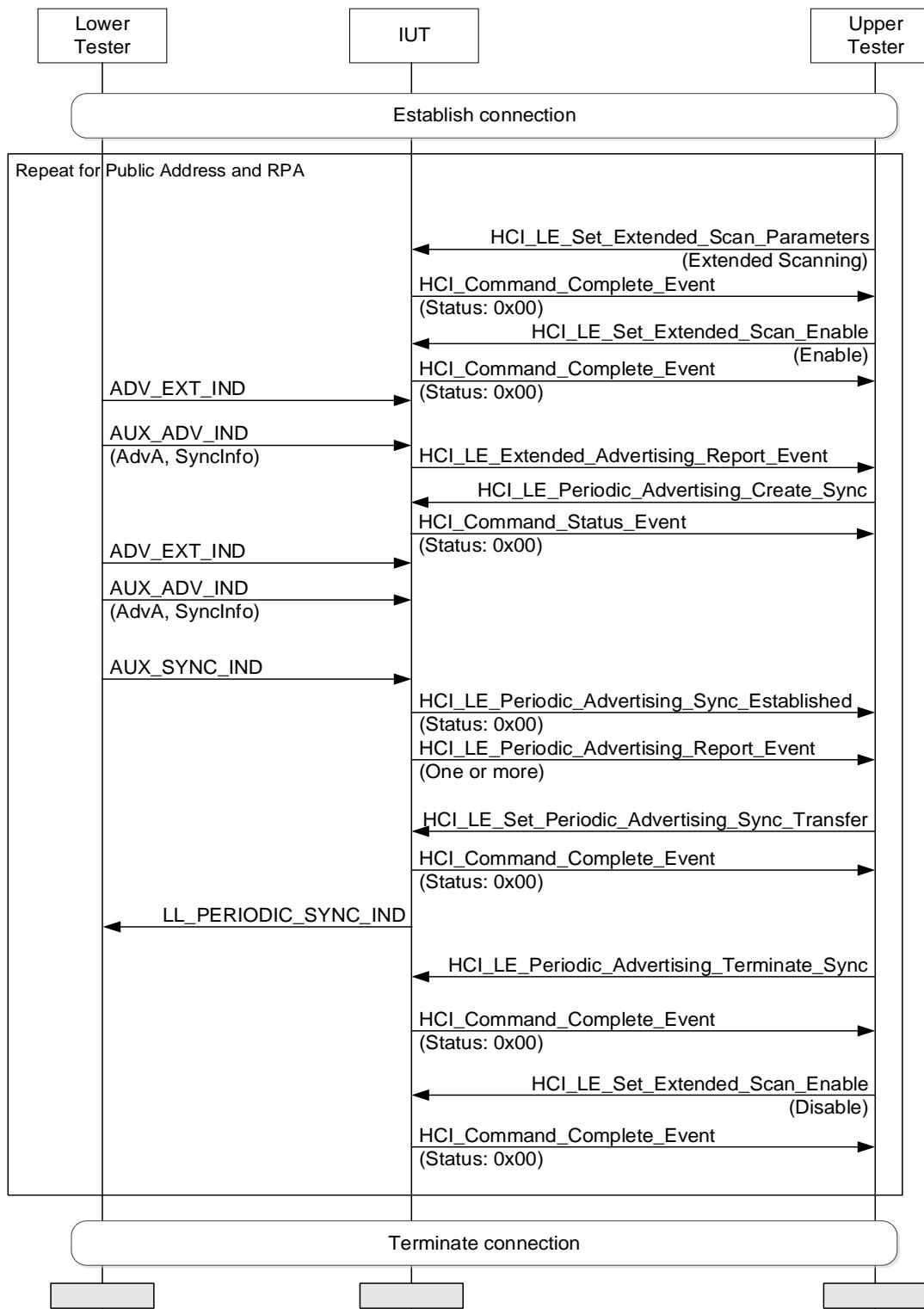


Figure 4.279: [Periodic Advertising Sync Transfer Procedure, Synchronized IUT Initiated]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.46](#), and the Lower Tester connects with the IUT on that PHY.

2. The Lower Tester begins extended advertising using the PHY indicated in the Primary ADV PHY column in [Table 4.46](#) for the primary advertising PHY, and the PHY indicated in the Secondary ADV PHY column in [Table 4.46](#) for the secondary advertising PHY, using a public address.
 3. The Lower Tester begins periodic advertising for the extended advertising in step 2.
 4. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Parameters command to the IUT with the Scanning_PHYs parameter set to the PHY indicated in the Primary ADV PHY column in [Table 4.46](#) and Scan_Type[0] set to 0x00 (Passive Scanning).
 5. The Upper Tester sends an HCI_LE_Set_Extended_Scan_Enable command to the IUT to enable scanning.
 6. The Upper Tester expects an HCI_LE_Extended_Advertising_Report event from the IUT containing a nonzero Periodic_Advertising_Interval.
 7. The Upper Tester sends an HCI_LE_Periodic_Advertising_Create_Sync command to the IUT to synchronize with the Lower Tester's periodic advertisements.
 8. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Established event from the IUT containing a Status of 0x00 (Success) and other fields matching the advertisements generated by the Lower Tester.
 9. The Upper Tester expects HCI_LE_Periodic_Advertising_Report events from the IUT.
 10. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer command to the IUT with the Sync_Handle of the Lower Tester's periodic advertising, Service_Data set to any non-zero value, and the Connection_Handle of the connection and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 11. The Lower Tester expects an LL_PERIODIC_SYNC_IND PDU from the IUT with ID set to the value used for Service_Data in step 10, SyncInfo and connEventCount set to valid values, SID set to match the advertising set ID representing the advertising in step 2, AType set to the address type used for the advertising in step 2, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in [Table 4.46](#), and the AdvA set to the Lower Tester's advertiser address used for the advertising in step 2.
 12. The Lower Tester verifies that the IUT continues maintaining the connection.
 13. The Lower Tester uses the SyncInfo and connEventCount values to verify that they allow synchronization to the periodic advertising from step 3.
 14. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate synchronization to periodic advertising.
 15. The Upper Tester sends an HCI_LE_Set_Extended_Scanning_Enable command to the IUT to disable scanning.
 16. The Lower Tester stops both extended advertising and periodic advertising.
 17. If the Test Privacy column in [Table 4.46](#) indicates No, then skip to step 19.
 18. If the IUT supports controller-based privacy, then repeat steps 2-16, except that in step 2 the Lower Tester uses a resolvable private address matching an IRK in the IUT's resolving list for the advertiser address, and that in step 11, the AdvA is set to the Lower Tester's unresolved advertising address used for the advertising in step 2.
 19. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.



4.3.6.4 [Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYs]

- Test Purpose

Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure for each combination of periodic advertising PHYs and connection PHYs.

For various PHYs, the Lower Tester transmits periodic advertising and transfers synchronization information for that periodic advertising, and the Upper Tester verifies that the IUT synchronizes successfully to that periodic advertising and reports that periodic advertising. Additionally, it is verified that the IUT ignores the Periodic Advertiser List.

- Reference

[13] 4.6.24

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.47](#) for each applicable Test Case ID.

Test Case ID	Connection PHY	Primary ADV PHY	Secondary ADV PHY	Test Periodic Advertiser List
	Step 1	Step 2	Step 2, Step 4	Step 8
LL/CON/SLA/BV-98-C LL/CON/MAS/BV-94-C	LE 1M	0x01 (LE 1M)	0x01 (LE 1M)	Yes
LL/CON/SLA/BV-99-C LL/CON/MAS/BV-95-C	LE 1M	0x03 (LE Coded)	0x03 (LE Coded)	No
LL/CON/SLA/BV-100-C LL/CON/MAS/BV-96-C	LE 1M	0x01 (LE 1M)	0x02 (LE 2M)	No
LL/CON/SLA/BV-101-C LL/CON/MAS/BV-97-C	LE Coded	0x01 (LE 1M)	0x01 (LE 1M)	No
LL/CON/SLA/BV-102-C LL/CON/MAS/BV-98-C	LE 2M	0x01 (LE 1M)	0x01 (LE 1M)	No

Table 4.47: Values for each Test Case ID



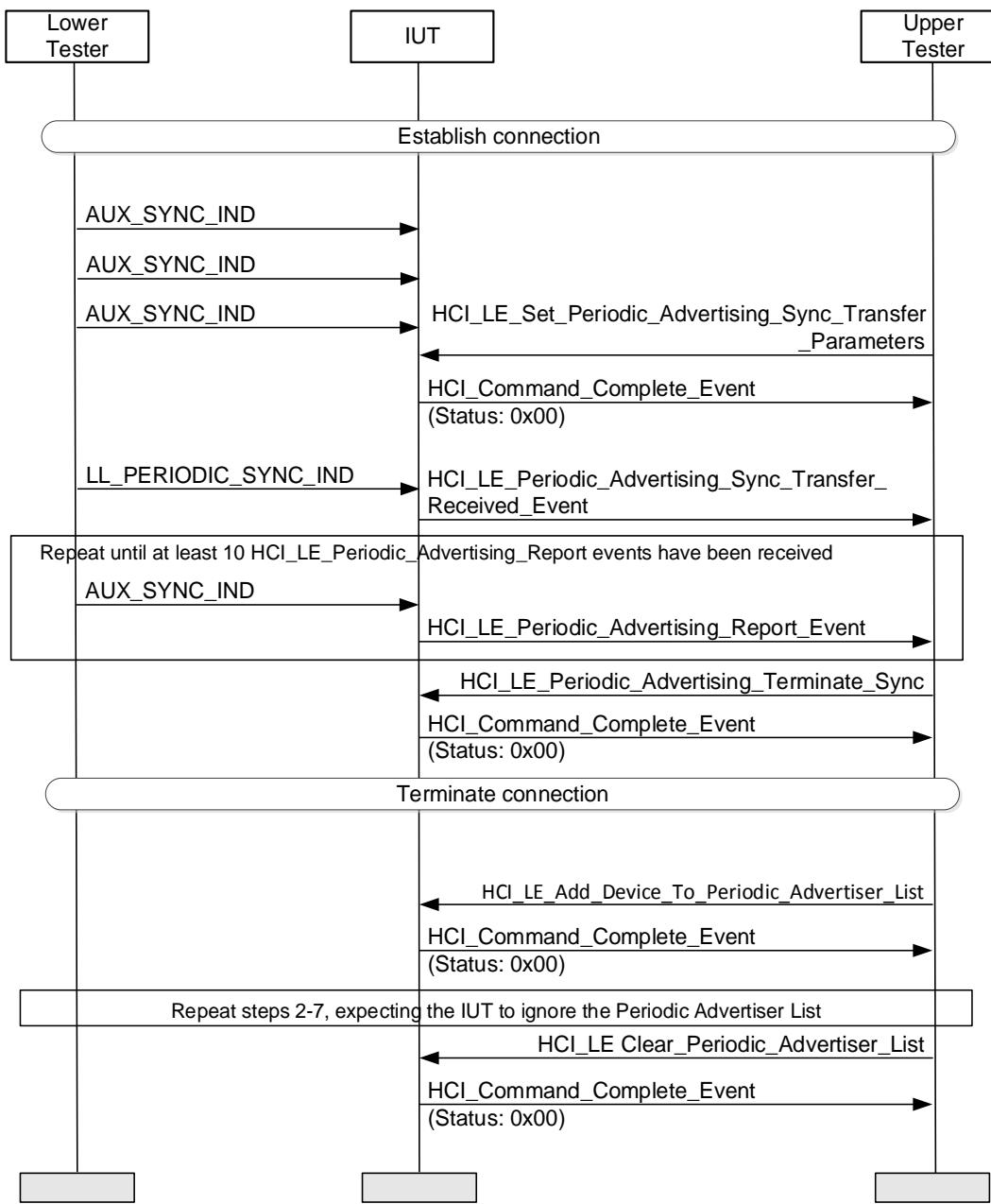


Figure 4.280: [Periodic Advertising Sync Transfer Procedure, Accepting – Different PHYs]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.47](#), and the Lower Tester connects with the IUT on that PHY.
 2. The Lower Tester begins periodic advertising using the PHY indicated in the Secondary ADV PHY column in [Table 4.47](#) and any periodic interval. The Lower Tester may begin extended advertising using the PHY indicated in the Primary ADV PHY column in [Table 4.47](#) for the primary advertising PHY, and the PHY indicated in the Secondary ADV PHY column in [Table 4.47](#) for the secondary advertising PHY, using a public address; if so, this refers to the periodic advertising.
 3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to

- 0, and Sync_Timeout set to 3 times the periodic advertising interval and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
4. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount and SyncInfo set to any values that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to the value indicated in the Secondary ADV PHY column in [Table 4.47](#), and the Adva set to the Lower Tester's advertiser address used for the advertising in step 2.
 5. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester, and the Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
 6. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
 7. The Lower Tester stops periodic advertising and extended advertising (if started).
 8. If the Test Periodic Advertiser List column in [Table 4.47](#) indicates No, then skip to step 12.
 9. The Upper Tester sends an HCI_LE_Add_Device_To_Periodic_Advertiser_List command to the IUT, containing an advertiser address different than the one used by the Lower Tester for periodic advertising, and expects an HCI_Command_Complete event in response.
 10. Repeat steps 2-7, expecting the IUT to ignore the Periodic Advertiser List.
 11. The Upper Tester sends an HCI_LE_Clear_Periodic_Advertiser_List command to the IUT and expects an HCI_Command_Complete event in response.
 12. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising, and generates reports after synchronization.

[Periodic Advertising Sync Transfer Procedure, Accepting – Skipping Events]

- Test Case IDs

[LL/CON/SLA/BV-103-C](#)

[LL/CON/MAS/BV-99-C](#)

- Test Purpose

Tests that the IUT is able to send reports for a periodic advertising after synchronization at least as often as allowed by the Skip value.

- Reference

[\[13\] 4.6.24](#)

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.



- Test Procedure

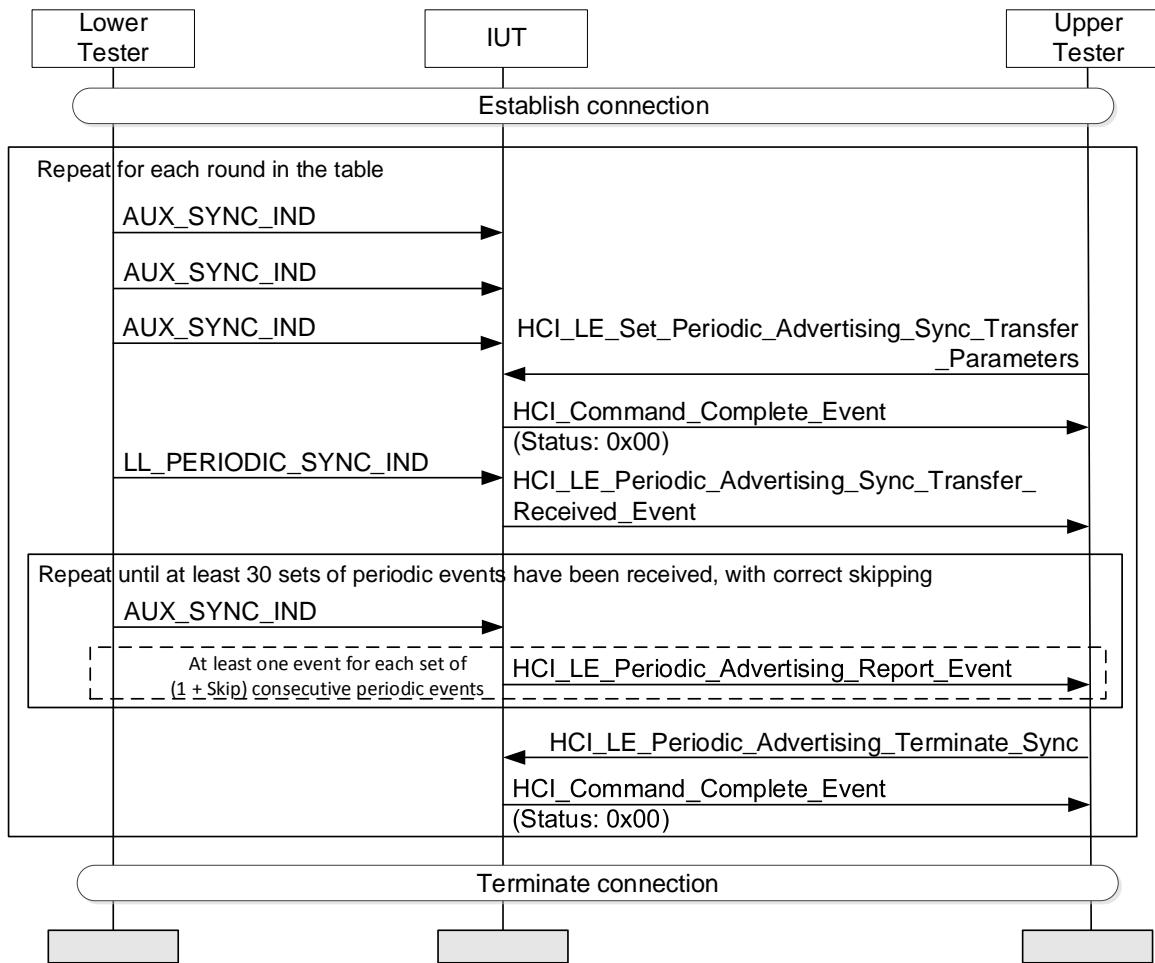


Figure 4.281: [Periodic Advertising Sync Transfer Procedure, Accepting – Skipping Events]

- The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
- Execute steps 3-9 for each round shown in [Table 4.48](#).

Round	Skip
	Step 4
1	0x0000
2	0x0001
3	0x0013

Table 4.48: Values for each case variation.

- The Lower Tester begins periodic advertising using the LE 1M PHY and a periodic interval greater than: (the max offset describable in a SyncInfo) + 2 x (the connection interval). The periodic advertising data for each periodic advertising event contains the event number, to allow the Upper Tester to determine exactly which periodic advertising events the IUT is reporting. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.



4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to a value as specified in [Table 4.48](#) for this round, and Sync_Timeout set to (Skip + 3) x the periodic advertising interval and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
5. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount set to the nearest connection event for which a valid offset in the SyncInfo can be specified to point to the periodic advertising, SyncInfo set to the value that would allow synchronization to the periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the Lower Tester's advertiser address used for the advertising in step 3.
6. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester.
7. The Upper Tester expects to receive at least one HCI_LE_Periodic_Advertising_Report event from the IUT for every (1 + Skip) periodic events; repeat until at least 30 sets of (1+Skip) periodic events have passed.
8. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
9. The Lower Tester stops periodic advertising and extended advertising (if started).
10. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT generates reports after synchronization at least as often as one report for every (1 + Skip) periodic events. The event numbers reported to the Upper Tester are no more than (1 + Skip) apart; two omissions out of the 30 repetitions are allowed, in order to account for packet loss.

4.3.6.6 [Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]

- Test Case IDs

[LL/CON/SLA/BV-104-C](#)

[LL/CON/MAS/BV-100-C](#)

- Test Purpose

Tests that the IUT synchronized to periodic advertising does not report synchronizing a second time if it receives synchronization information for the same periodic advertising.

- Reference

[\[13\] 4.6.24](#)

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.



- Test Procedure

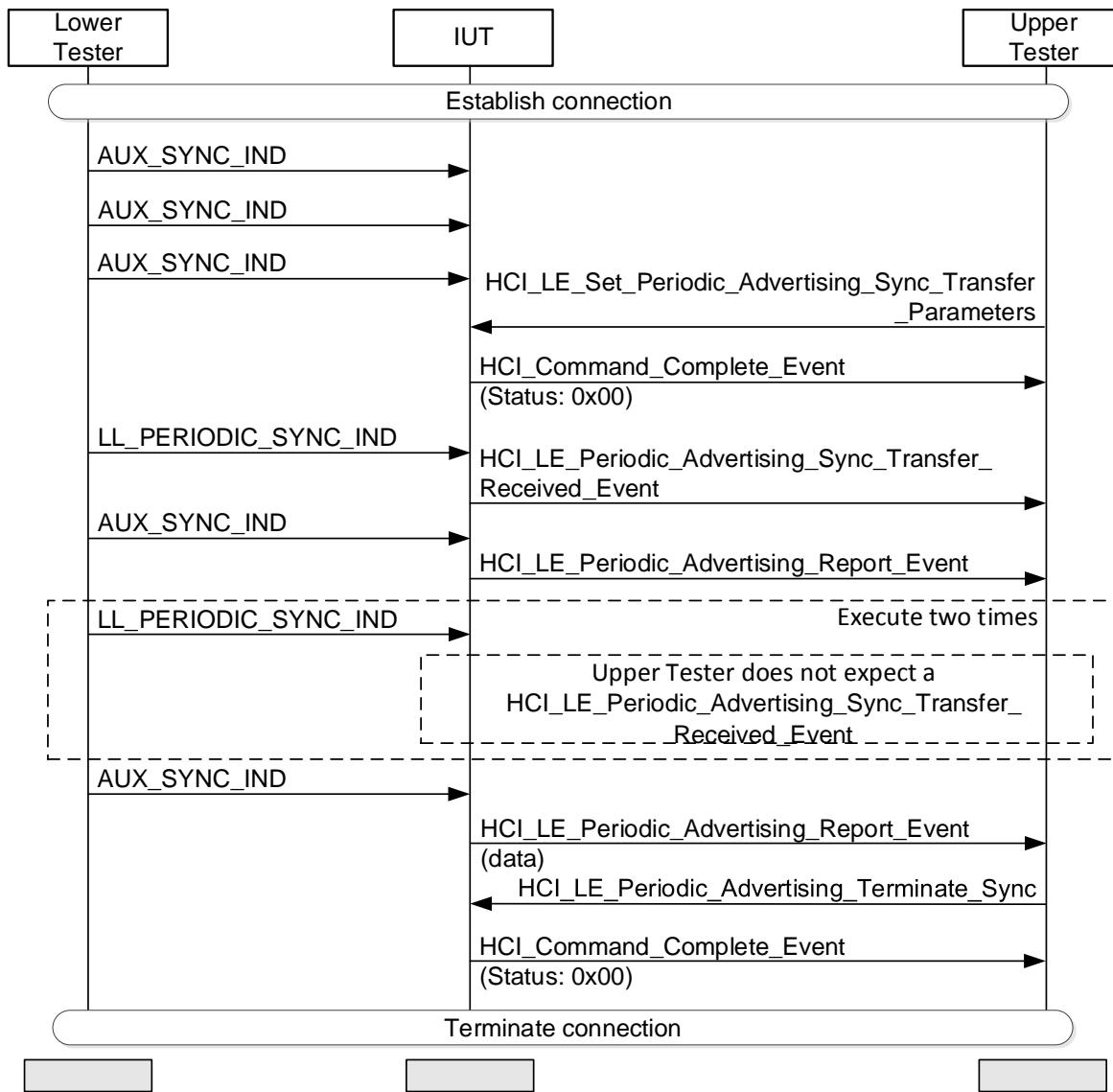


Figure 4.282: [Periodic Advertising Sync Transfer Procedure, Accepting – Already Synchronized]

- The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
- The Lower Tester begins periodic advertising using the LE 1M PHY and any periodic interval. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.
- The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
- The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, connEventCount and SyncInfo set to any values that would allow synchronization to the periodic advertising, SID set any valid value, ATYPE set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the Lower Tester's advertiser address used for the advertising in step 2.



5. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU.
6. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
7. Repeat step 4 two more times, with parameters describing the same periodic advertising. On each execution of this step, the Upper Tester expects not to receive any additional HCI_LE_Periodic_Advertising_Sync_Transfer_Received event.
8. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester expects at least 10 more HCI_LE_Periodic_Advertising_Report events from the IUT.
9. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
10. The Lower Tester stops periodic advertising and extended advertising (if started).
11. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: The IUT does not generate a second HCI_LE_Periodic_Advertising_Sync_Transfer_Received event for the same periodic advertising, but continues to generate HCI_LE_Periodic_Advertising_Report events with the same Sync_Handle.

4.3.6.7 [Periodic Advertising Sync Transfer Procedure, Accepting – Extreme Timings]

- Test Case IDs

[LL/CON/SLA/BV-105-C](#)

[LL/CON/MAS/BV-101-C](#)

- Test Purpose

Tests that the IUT is able to synchronize to periodic advertising using connEventCount values that represent short and large distances into the past and future.

- Reference

[\[13\] 4.6.24](#)

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.



- Test Procedure

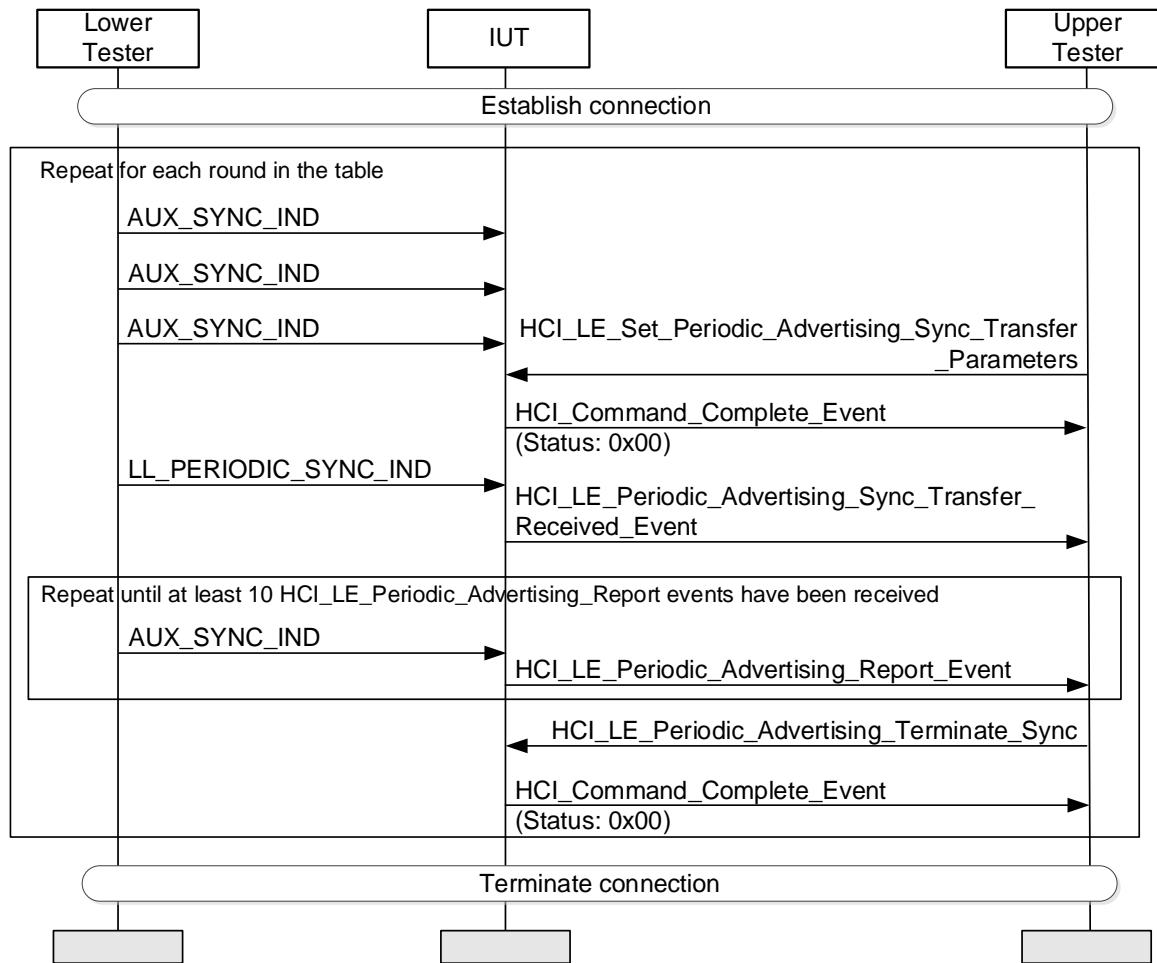


Figure 4.283: [Periodic Advertising Sync Transfer Procedure, Accepting – Extreme Timings]

- The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
- Execute steps 3-9 for each round shown in [Table 4.49](#).

Round	Lower Tester's Periodic Advertising Interval	connEventCount
	Step 3	Step 5
1	greater than (the max offset describable in a SyncInfo) + 2 x connection interval	Minimum distance in the future
2	greater than (the max offset describable in a SyncInfo) + 2 x connection interval	Maximum distance in the future
3	greater than (the max offset describable in a SyncInfo) + 2 x connection interval	Minimum distance in the past
4	greater than (the max offset describable in a SyncInfo) + 2 x connection interval	Maximum distance in the past
5	less than half the connection interval	Maximum distance in the past
6	less than half the connection interval	Minimum distance in the future

Table 4.49: Values for each case variation.



3. The Lower Tester begins periodic advertising using the LE 1M PHY and a periodic interval as specified in [Table 4.49](#). The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.
4. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to 3 times the periodic advertising interval and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
5. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values as specified in [Table 4.49](#) for this round that would allow synchronization to its periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address. In [Table 4.49](#), “Minimum distance” means the nearest connection event, and “Maximum distance” means the furthest connection event, in the specified direction for which a valid offset in the SyncInfo can be specified to point to the periodic advertising.
6. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU.
7. The Lower Tester continues to send periodic advertisements to the IUT. The Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
8. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
9. The Lower Tester stops periodic advertising and extended advertising (if started).
10. Terminate the connection between the IUT and the Lower Tester.

- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising and generates reports after synchronization.

[4.3.6.8 \[Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure\]](#)

- Test Case IDs

[LL/CON/SLA/BV-106-C](#)

[LL/CON/MAS/BV-102-C](#)

- Test Purpose

Tests that the IUT is able to report synchronization failure in a timely manner.

- Reference

[\[13\] 4.6.24](#)

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with an IRK in the resolving list for resolving a resolvable private address used in advertising from the Lower Tester.



- Test Procedure

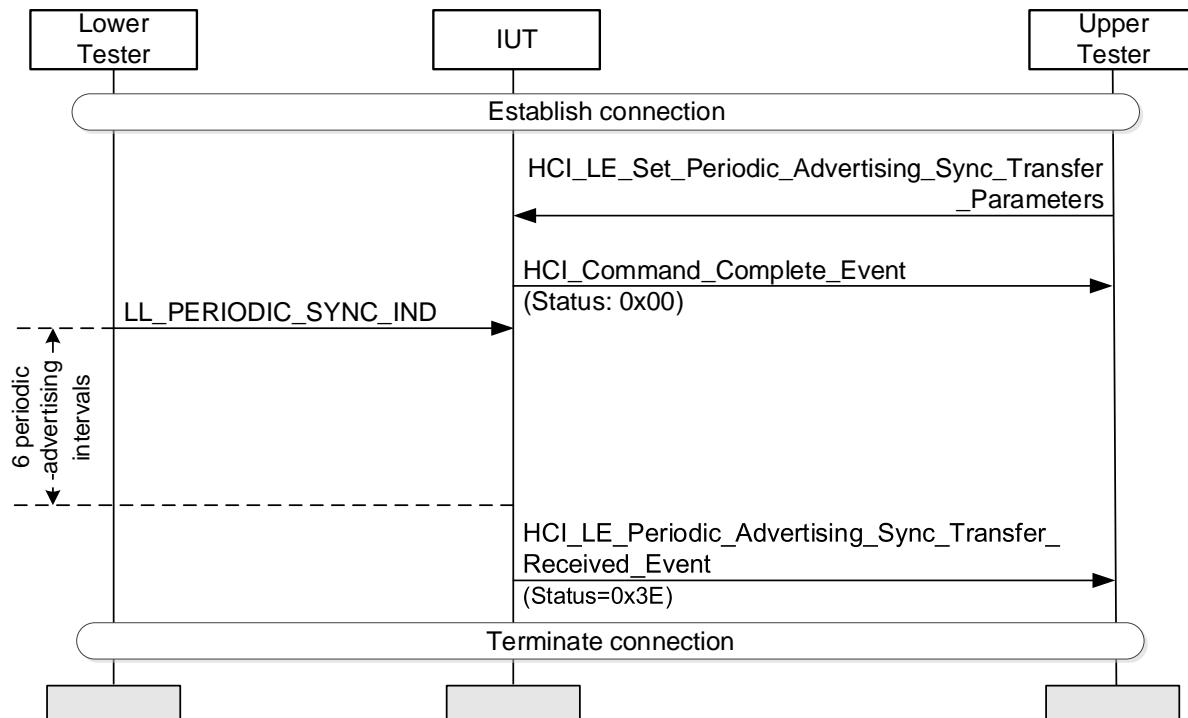


Figure 4.284: [Periodic Advertising Sync Transfer Procedure, Accepting – Synchronization Failure]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY. The periodic advertising interval should be around three times the connection interval.
 2. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, Skip set to 0, and Sync_Timeout set to at least 12 periodic advertising intervals, and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 3. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address.
 4. After 6 periodic advertising intervals have passed, starting with the moment where the next periodic event would be expected, the Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT with Status set to Connection Failed to be Established / Synchronization Timeout (0x3E).
 5. From the 7th periodic advertising event onwards, the Lower Tester sends AUX_SYNC_IND PDUs to the IUT at times corresponding to the data in the LL_PERIODIC_SYNC_IND PDU.
- Expected Outcome

Pass Verdict

MAS/SLA: The IUT reports a failure to synchronize after it does not receive any periodic advertising packet after 6 periodic advertising intervals.



Fail Verdict

MAS/SLA: The IUT sends an HCI_LE_Periodic_Advertising_Report event to the Upper Tester.

4.3.6.9 [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses]

- Test Case IDs

[LL/CON/SLA/BV-107-C](#)

[LL/CON/MAS/BV-103-C](#)

- Test Purpose

Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure for each Periodic Advertising Sync Transfer Mode. Public addresses are tested. If the IUT supports controller-based privacy, then Resolvable Private addresses are tested as well.

The Lower Tester transmits periodic advertising and transfers synchronization information for that periodic advertising, and the Upper Tester verifies that the IUT synchronizes (when enabled) successfully to that periodic advertising and reports (when enabled) that periodic advertising. For transfer modes including synchronization, enabling and disabling of reporting is tested. It is tested that the IUT uses the configured transfer mode for the connection instead of the default transfer mode is tested.

- Reference

[\[13\] 4.6.24](#)

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with an IRK in the resolving list for resolving a resolvable private address used in advertising from the Lower Tester.

The IUT shall not have performed the HCI_LE_Set_Default_Periodic_Advertising_Sync_Transfer_Parameters command since last being reset.



- Test Procedure

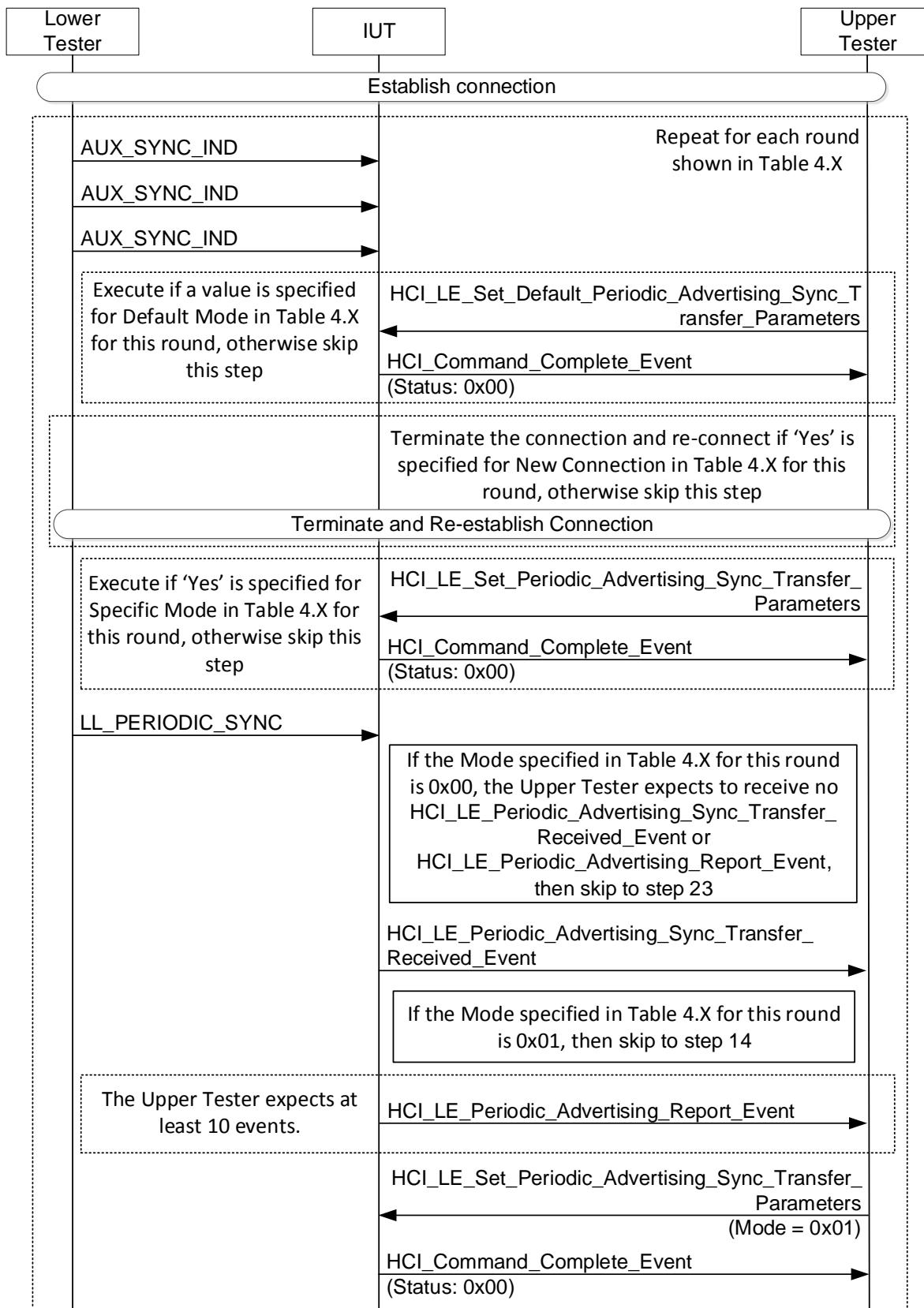


Figure 4.285: [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses] part 1 of 2



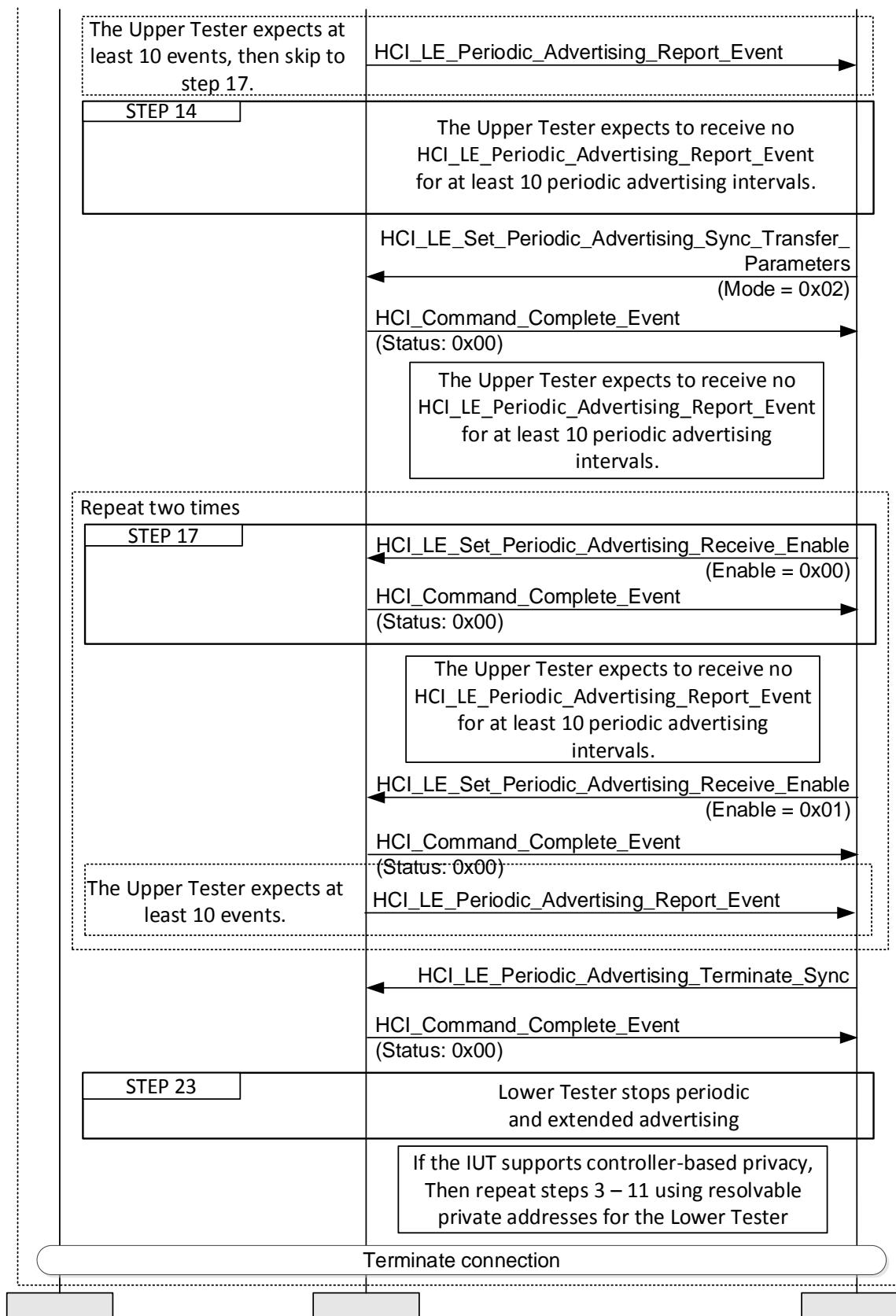


Figure 4.286: [Periodic Advertising Sync Transfer Procedure, Accepting – Different Modes and Addresses] part 2 of 2



1. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
2. Execute steps 3-23 for each round shown in [Table 4.50](#).

Round	Default Mode	New connection	Specific Mode	Mode
	Step 4	Step 5	Step 6	Steps 8 & 10
1	-	No	No	0x00
2	0x02	No	No	0x00
3	-	Yes	No	0x02
4	-	No	Yes	0x00
5	-	No	Yes	0x01
6	-	No	Yes	0x02

Table 4.50: Values for each case variation.

3. The Lower Tester begins periodic advertising using the LE 1M PHY and any periodic interval. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising. The Lower Tester uses a different advertising set ID for each round.
4. If a value is specified for Default Mode in [Table 4.50](#) for this round, the Upper Tester sends an HCI_LE_Set_Default_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to the value specified for Default Mode in [Table 4.50](#) for this round, Skip set to 0, and Sync_Timeout set to 30 times the periodic advertising interval set in step 3, and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
5. If "Yes" is specified for New Connection in [Table 4.50](#) for this round, terminate the connection between the IUT and the Lower Tester and then re-create it as in step 1.
6. If "Yes" is specified for Specific Mode in [Table 4.50](#) for this round, the Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to the value specified for Mode in [Table 4.50](#) for this round, Skip set to 0, and Sync_Timeout set to 30 times the periodic advertising interval, and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
7. The Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values that would allow synchronization to its periodic advertising, SID set to the value used in step 4, ATType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to the address used for the extended advertising in step 3 (or any valid public address if the Lower Tester is not advertising)..
8. If the Mode specified in [Table 4.50](#) for this round is 0x00 (no synchronization), the Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Sync_Transfer_Received or HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising interval, then skip to step 23.
9. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the LL_PERIODIC_SYNC_IND PDU and periodic advertisements generated by the Lower Tester.
10. If the Mode specified in [Table 4.50](#) for this round is 0x01 (synchronize with reporting initially disabled), skip to step 14.
11. The Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
12. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x01, and all



- other parameters set to the values specified in step 6, and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
13. The Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT, then skip to step 17.
 14. The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
 15. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with the connection handle of the connection, Mode set to 0x02, and all other parameters set to the values specified in step 6, and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 16. The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
 17. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x00 and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 18. The Upper Tester expects to receive no further HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
 19. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Receive_Enable command to the IUT with Enable set to 0x01 and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 20. The Upper Tester expects at least 10 HCI_LE_Periodic_Advertising_Report events from the IUT.
 21. Repeat steps 17 to 20.
 22. The Upper Tester sends an HCI_LE_Periodic_Advertising_Terminate_Sync command to the IUT to terminate periodic advertising reception and expects an HCI_Command_Complete event in response.
 23. The Lower Tester stops periodic advertising and extended advertising (if started).
 24. If the IUT supports controller-based privacy, then repeat steps 3-11 using the values for round 6 in [Table 4.50](#), except that in step 3 the Lower Tester uses a resolvable private address corresponding to a public identity address and matching an IRK in the IUT's resolving list for the advertiser address in the extended advertising, if started, that in step 7 the AdvA is set to the Lower Tester's unresolved advertising address, and that in step 9 the HCI_LE_Periodic_Advertising_Sync_Transfer_Received event the Advertiser_Address_Type is set to 0x02 (Public Identity Address (corresponds to Resolved Private Address)) and the Advertiser_Address is set to the Lower Tester's resolved advertiser address.
 25. If the IUT supports controller-based privacy, then repeat steps 3-11 using the values for round 6 in [Table 4.50](#), except that in step 3 the Lower Tester uses a resolvable private address not matching an IRK in the IUT's resolving list for the advertiser address in the extended advertising, if started, that in step 7 the AdvA is set to the Lower Tester's unresolved advertising address, and that in step 9 the HCI_LE_Periodic_Advertising_Sync_Transfer_Received event the Advertiser_Address_Type is set to 0x01 (Random Device Address / Random (static) Identity Address) and the Advertiser_Address is set to the Lower Tester's unresolved advertiser address.
 26. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT accepts periodic advertising sync information, synchronizes to periodic advertising only when configured by the transfer mode, and generates reports after synchronization only when configured by the transfer mode or when reporting is enabled.



4.3.6.10 [Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]

- Test Case IDs

LL/CON/SLA/BV-108-C

LL/CON/MAS/BV-104-C

- Test Purpose

Tests that the IUT is able to accept the Periodic Advertising Sync Transfer Procedure and synchronize successfully when the Transfer Mode is changed while synchronization is in progress.

- Reference

[13] 4.6.24

- Initial Condition

MAS/SLA: The IUT is not scanning, synchronized to periodic advertising, or maintaining any connections.

- Test Procedure

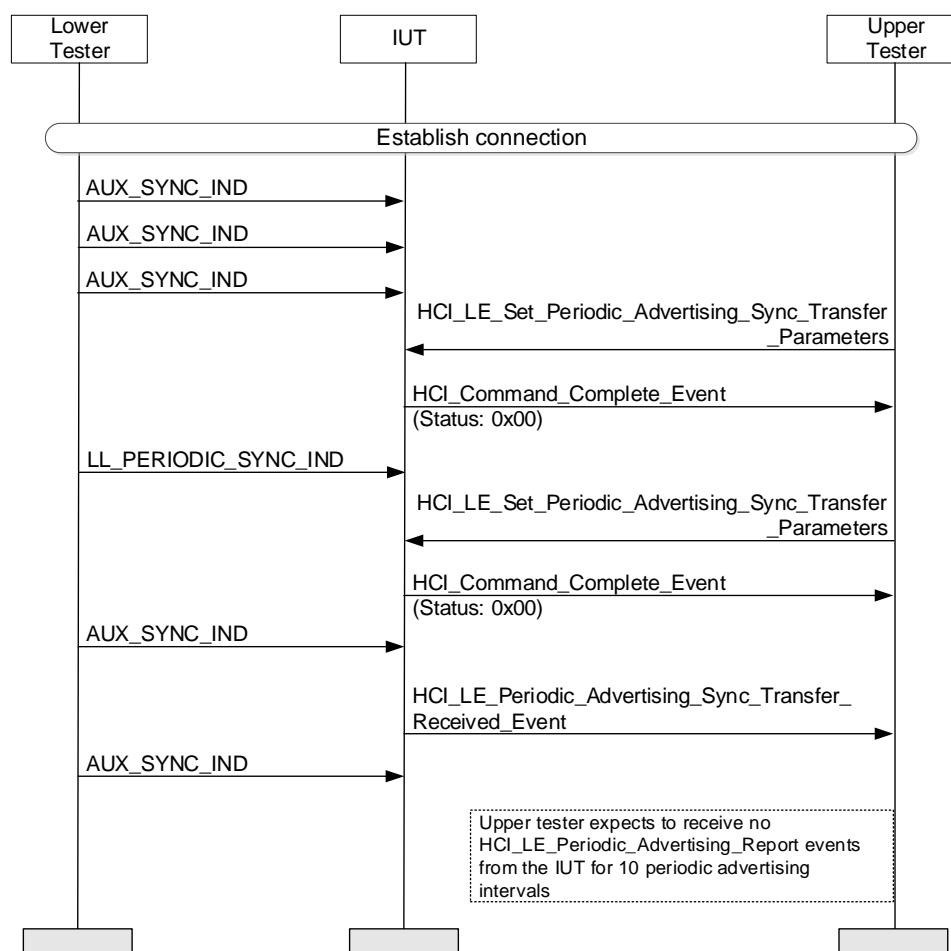


Figure 4.287: [Periodic Advertising Sync Transfer Procedure, Accepting, Changing Transfer Mode during Synchronization]



1. The Lower Tester begins periodic advertising using the LE 1M PHY with a periodic advertising interval large enough to support the timing requirements in steps 4-6. The Lower Tester may begin extended advertising using the LE 1M PHY for the primary advertising PHY and the LE 1M PHY for the secondary advertising PHY using a public address; if so, this refers to the periodic advertising.
 2. The Upper Tester configures the IUT to connect with the Lower Tester on the LE 1M PHY, and the Lower Tester connects with the IUT on the LE 1M PHY.
 3. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to 0x01 (synchronize with reporting initially disabled), Skip set to 0, and Sync_Timeout set to any value and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 4. At a time when the next periodic advertising event would occur far enough in the future to allow completing this step and step 6 before that periodic advertising event is started, the Lower Tester sends an LL_PERIODIC_SYNC_IND PDU to the IUT with ID set to any non-zero value, SyncInfo and connEventCount set to valid values that would allow synchronization to its periodic advertising, SID set any valid value, AType set to the public address type, RFU set to 000, the PHY set to LE 1M, and the AdvA set to any valid public address. The Lower Tester expects a LL data PDU from the IUT, acknowledging the receipt of the LL_PERIODIC_SYNC_IND PDU.
 5. The Upper Tester sends an HCI_LE_Set_Periodic_Advertising_Sync_Transfer_Parameters command to the IUT with Mode set to 0x02 (synchronize with reporting initially enabled), Skip set to 0, and Sync_Timeout set to any value and expects an HCI_Command_Complete with Status set to 0x00 (Success) in response.
 6. The Upper Tester expects an HCI_LE_Periodic_Advertising_Sync_Transfer_Received event from the IUT containing a Status of 0x00 (Success), the connection handle of the connection, and other fields matching the advertisements generated by the Lower Tester.
 7. The Upper Tester expects to receive no HCI_LE_Periodic_Advertising_Report events from the IUT for at least 10 periodic advertising intervals.
- Expected Outcome

Pass Verdict

MAS/SLA: The IUT accepts periodic advertising sync information and synchronizes to periodic advertising following the transfer mode used during reception of the periodic advertising sync information.

4.3.6.11 [Acknowledging Long Control PDUs]

- Test Purpose

Tests that the IUT is able to receive a long LL Control PDU, acknowledge it, and continue maintaining the connection. The test purpose is exercised over each supported PHY.

- Reference

[13] 4.5.11

- Initial Condition

MAS/SLA: The IUT is not advertising or maintaining any connections. If the IUT supports controller-based privacy, then the IUT has been configured with a local IRK in the resolving list for advertising to the Lower Tester with a resolvable private address.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.51](#) for each applicable Test Case ID.



Test Case ID	Connection PHY
	Step 1
LL/CON/SLA/BV-109-C	LE 1M
LL/CON/MAS/BV-105-C	
LL/CON/SLA/BV-110-C	LE 2M
LL/CON/MAS/BV-106-C	
LL/CON/SLA/BV-111-C	LE Coded
LL/CON/MAS/BV-107-C	

Table 4.51: Values for each Test Case ID

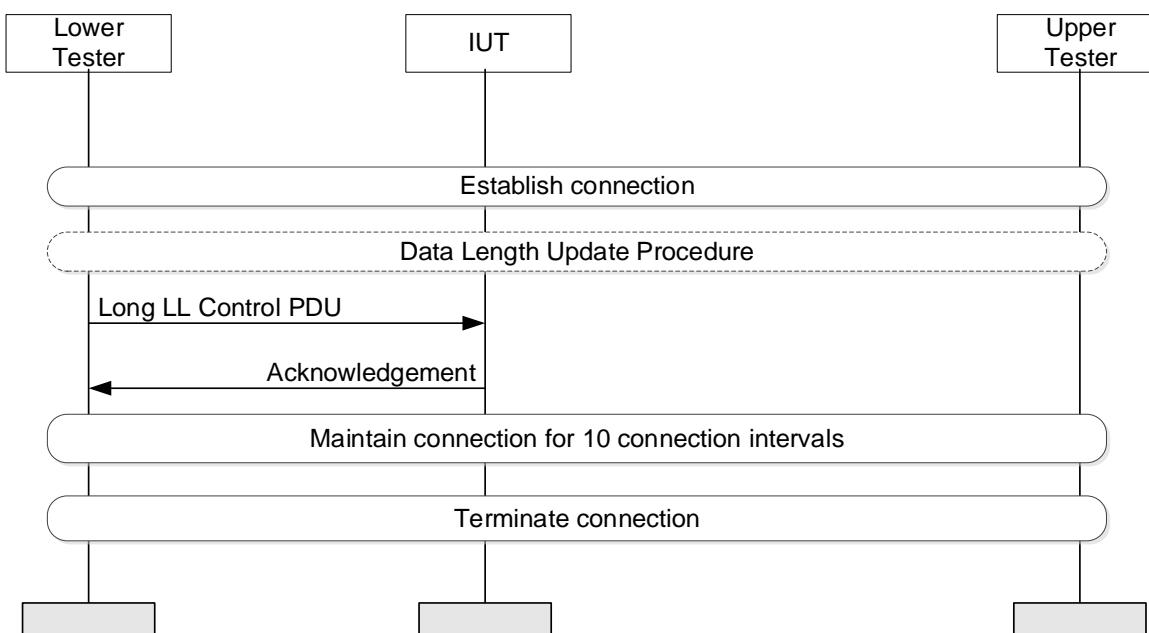


Figure 4.288: [Acknowledging Long Control PDUs]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.51](#), and the Lower Tester connects with the IUT on that PHY.
 2. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum value (27 octets).
 3. The Lower Tester transmits any LL Control PDU of a length supported by the IUT for reception that is greater than 27 octets.
 4. The Lower Tester expects the IUT to send an LL acknowledgement and to continue maintaining the connection for at least 10 connection events.
 5. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: In each case, the IUT transfers synchronization information matching the periodic advertising and continues maintaining the connection.



- Notes

To determine the maximum LL Control PDU length supported by the IUT (for a received PDU), a value that is required in step 3 of the test procedure, the tester may perform an LL feature exchange procedure or it may use the information available in the ICS.

4.3.6.12 [Rejecting Request to Send Long Control PDUs before Feature Exchange]

- Test Purpose

Tests that the IUT does not send a long LL Control PDU before feature exchange, including when limits set by Data PDU Length Management allow sending LL Data PDUs of equal or greater length (if supported). The test purpose is exercised over each supported PHY.

- Reference

[13] 4.5.11

- Initial Condition

MAS/SLA: The IUT is not maintaining any connections.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.52](#) for each applicable Test Case ID.

Test Case ID	Connection PHY
	Step 1
LL/CON/SLA/BV-112-C LL/CON/MAS/BV-108-C	LE 1M
LL/CON/SLA/BV-113-C LL/CON/MAS/BV-109-C	LE 2M
LL/CON/SLA/BV-114-C LL/CON/MAS/BV-110-C	LE Coded

Table 4.52: Values for each Test Case ID



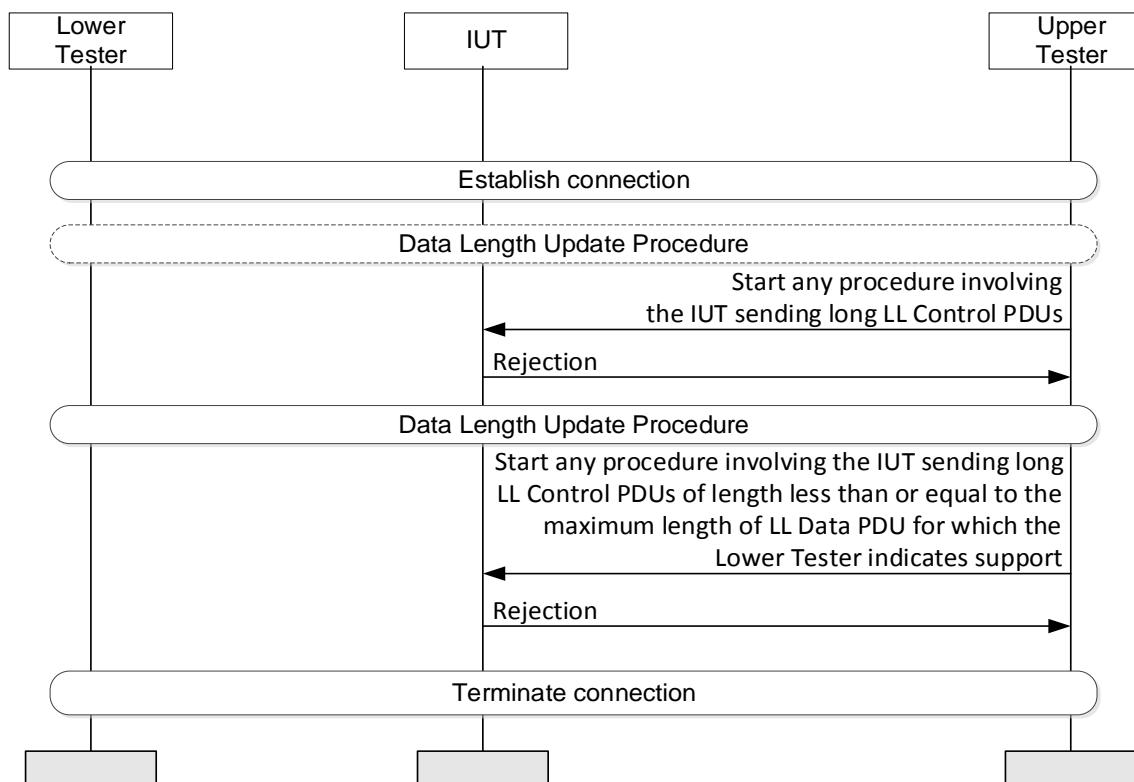


Figure 4.289: [Rejecting Request to Send Long Control PDUs before Feature Exchange]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.52](#), and the Lower Tester connects with the IUT on that PHY.
 2. If the IUT autonomously initiates feature exchange at any time between step 1 and step 9, the test ends with an Inconclusive Verdict.
 3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum values (27 octets and 328 us).
 4. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU (containing a CtrData field of 27 octets or larger).
 5. The Upper Tester expects the IUT to reject the command.
 6. If the IUT does not support the Data Length Update Procedure, skip steps 7-9.
 7. The Lower Tester initiates the Data Length Update Procedure with the maximum values and expects the IUT to complete the procedure.
 8. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU of length equal to or less than the maximum length of LL Data PDU that the Lower Tester indicated support for reception in the previous step.
 9. The Upper Tester expects the IUT to reject the command.
 10. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: The IUT rejects commands to initiate procedures involving long LL Control PDUs before feature exchange.



4.3.6.13 [Rejecting Request to Send Long Control PDUs after Feature Exchange]

- Test Purpose

Tests that the IUT does not send a long LL Control PDU after feature exchange indicating lack of support of any features using LL Control PDUs of that length or greater by the Lower Tester, including when limits set by Data PDU Length Management allow sending LL Data PDUs of equal or greater length (if supported). The test purpose is exercised over each supported PHY.

- Reference

[13] 4.5.11

- Initial Condition

MAS/SLA: The IUT is not maintaining any connections.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.53](#) for each applicable Test Case ID.

Test Case ID	Connection PHY
	Step 1
LL/CON/SLA/BV-115-C	LE 1M
LL/CON/MAS/BV-111-C	
LL/CON/SLA/BV-116-C	LE 2M
LL/CON/MAS/BV-112-C	
LL/CON/SLA/BV-117-C	LE Coded
LL/CON/MAS/BV-113-C	

Table 4.53: Values for each Test Case ID



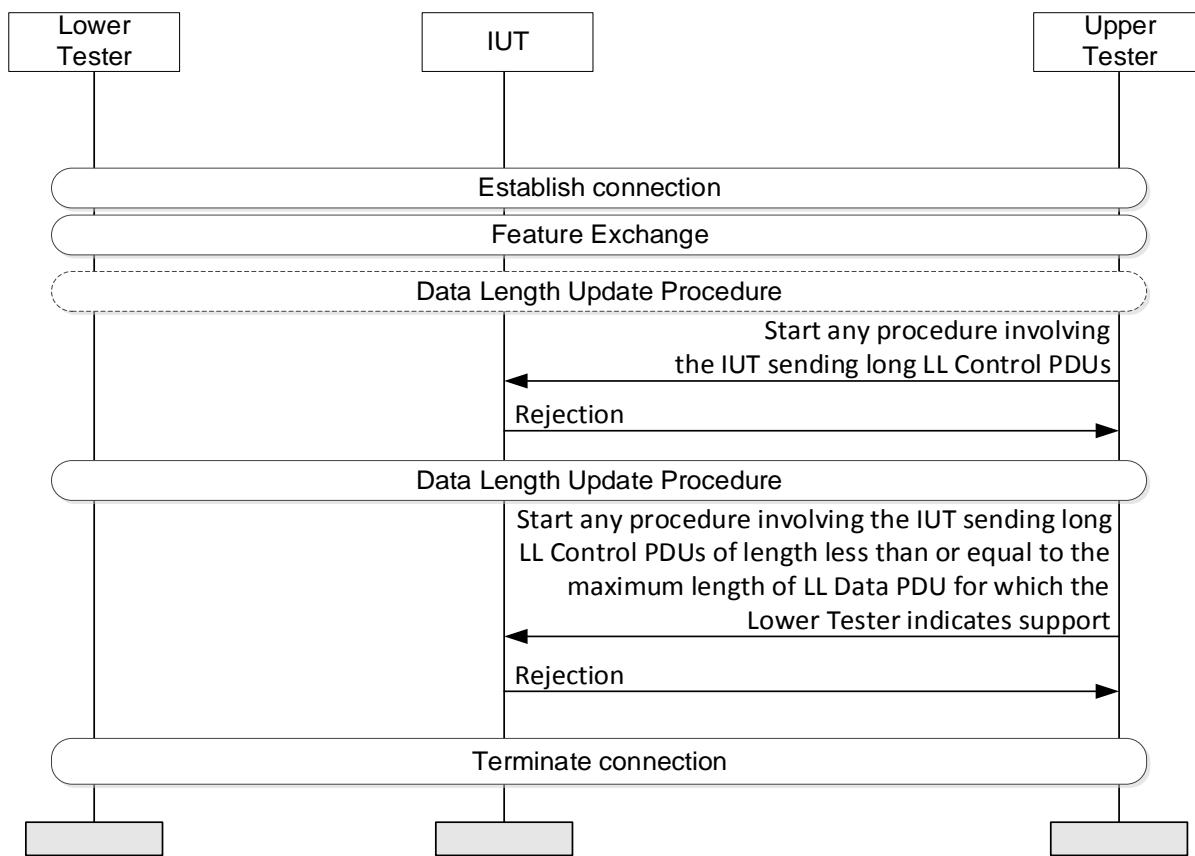


Figure 4.290: [Rejecting Request to Send Long Control PDUs after Feature Exchange]

1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.53](#), and the Lower Tester connects with the IUT on that PHY.
 2. The Lower Tester initiates feature exchange or responds to feature exchange, if autonomously initiated by the IUT, indicating no support for any procedures involving long LL Control PDUs.
 3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum values (27 octets and 328 us).
 4. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU (containing a CtrData field of 27 octets or larger).
 5. The Upper Tester expects the IUT to reject the command.
 6. If the IUT does not support the Data Length Update Procedure, skip steps 7-9.
 7. The Lower Tester initiates the Data Length Update Procedure with the maximum values and expects the IUT to complete the procedure.
 8. The Upper Tester commands the IUT to initiate any supported procedure that involves the IUT transmitting a long LL Control PDU of length equal to or less than the maximum length of LL Data PDU that the Lower Tester indicated support for reception in the previous step.
 9. The Upper Tester expects the IUT to reject the command with the error code 0x1A (Unsupported Remote Feature).
 10. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: The IUT rejects commands to initiate procedures involving long LL Control PDUs after feature exchange indicating that the Lower Tester does not support any such procedures.



4.3.6.14 [Sending Long Control PDUs after Feature Exchange]

- Test Purpose

Tests that the IUT is able to send a long LL Control PDU after feature exchange indicating support of any features using LL Control PDUs of that length or greater by the Lower Tester, including when limits set by Data PDU Length Management disallow sending LL Data PDUs of that length, and continue maintaining the connection. The test purpose is exercised over each supported PHY.

- Reference

[13] 4.5.11

- Initial Condition

MAS/SLA: The IUT is not maintaining any connections.

- Test Procedure and Test Case IDs

Execute the test procedure using the parameters indicated in [Table 4.54](#) for each applicable Test Case ID.

Test Case ID	Connection PHY
	Step 1
LL/CON/SLA/BV-118-C	LE 1M
LL/CON/MAS/BV-114-C	
LL/CON/SLA/BV-119-C	LE 2M
LL/CON/MAS/BV-115-C	
LL/CON/SLA/BV-120-C	LE Coded
LL/CON/MAS/BV-116-C	

Table 4.54: Values for each Test Case ID

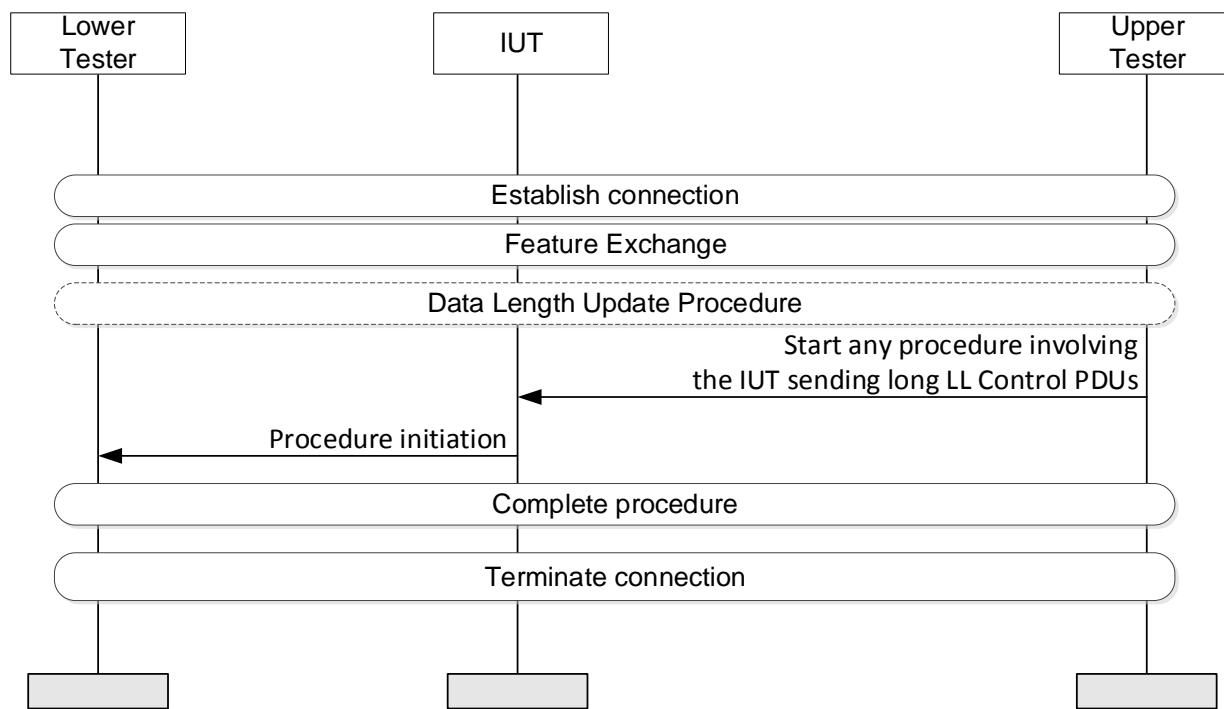


Figure 4.291: [Sending Long Control PDUs after Feature Exchange]



1. The Upper Tester configures the IUT to connect with the Lower Tester on the PHY indicated in the Connection PHY column in [Table 4.54](#), and the Lower Tester connects with the IUT on that PHY.
 2. The Lower Tester initiates feature exchange or responds to feature exchange, if autonomously initiated by the IUT, indicating support for at least one procedure involving long LL Control PDUs that the IUT supports.
 3. If the IUT autonomously initiates the Data Length Update Procedure, the Lower Tester responds indicating it supports the minimum value (27 octets).
 4. The Upper Tester commands the IUT to initiate any procedure for which the Lower Tester indicated support that involves the IUT transmitting a long LL Control PDU (i.e., payload length larger than 27 octets).
 5. The Lower Tester expects the IUT to initiate the procedure.
 6. The Lower Tester continues the procedure at least until the IUT has sent the expected, valid long PDU.
 7. Terminate the connection between the IUT and the Lower Tester.
- Expected Outcome

Pass Verdict

MAS/SLA: The IUT sends long LL Control PDUs after feature exchange indicating that the Lower Tester supports them.

[4.3.6.15 \[Constant Tone Extension Request Procedure, IUT Initiated, Unsupported\]](#)

- Test Purpose

Tests that the IUT correctly handles the case where the remote does not support the Connection CTE Response feature.

The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and confirms the IUT correctly handles the LL_UNKOWN_RSP PDU generated by the Lower Tester. When aware of the missing feature support on the remote Lower Tester, the IUT will not initiate a CTE Request Procedure.

- Test Case IDs

[LL/CON/SLA/BV-63-C](#)

[LL/CON/MAS/BV-59-C](#)

- Reference

[\[10\]](#) 5.1.12

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

Execute the test procedure for IUT as Master (if supported) and IUT as Slave (if supported). If the Lower Tester has exchanged features with the IUT, skip steps 2-5.



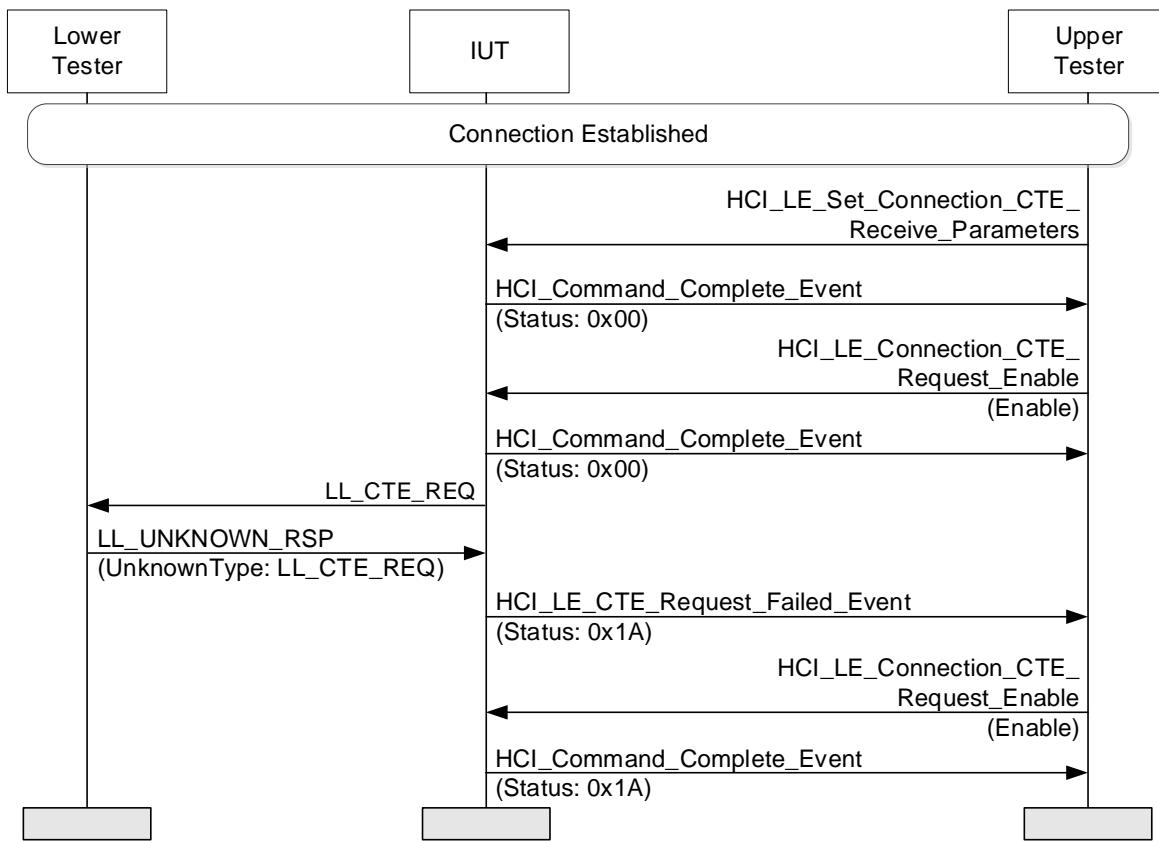


Figure 4.292: [Constant Tone Extension Request Procedure, IUT Initiated, Unsupported]

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
 2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).
 3. The Lower Tester expects an LL_CTE_REQ PDU.
 4. The Lower Tester sends an LL_UNKNOWN_RSP PDU. UnknownType is set to LL_CTE_REQ.
 5. The Upper Tester expects an HCI_LE_CTE_Request_Failed event from the IUT. Status shall be set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).
 6. Repeat step 2.
- Expected Outcome

Pass Verdict

On step 2, the IUT sends an LL_CTE_REQ PDU. The Upper Tester expects an HCI_LE_CTE_Request_Failed event with the Status set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).



On step 6, The IUT does not send any LL_CTE_REQ PDU. The Upper Tester expects an HCI_Command_Complete event with the Status set to 0x1A (Unsupported Remote Feature / Unsupported LMP Feature).

4.3.6.16 [Unrequested Constant Tone Extension, IUT Receiving, AoD]

- Test Purpose

Tests that an IUT can process an unrequested AoD Constant Tone Extension from a remote Lower Tester. Test that the IUT sends the appropriate event to the Upper Tester based on the Lower Tester's Constant Tone Extension.

The Lower Tester maintains a connection and transmits an AoD Constant Tone Extension on a data packet. The Upper Tester observes the event returned by the IUT.

- Test Case IDs

[LL/CON/SLA/BV-70-C](#)

[LL/CON/MAS/BV-66-C](#)

- Reference

[\[13\] 2.5.4, 5.1.12](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value)



- Test Procedure

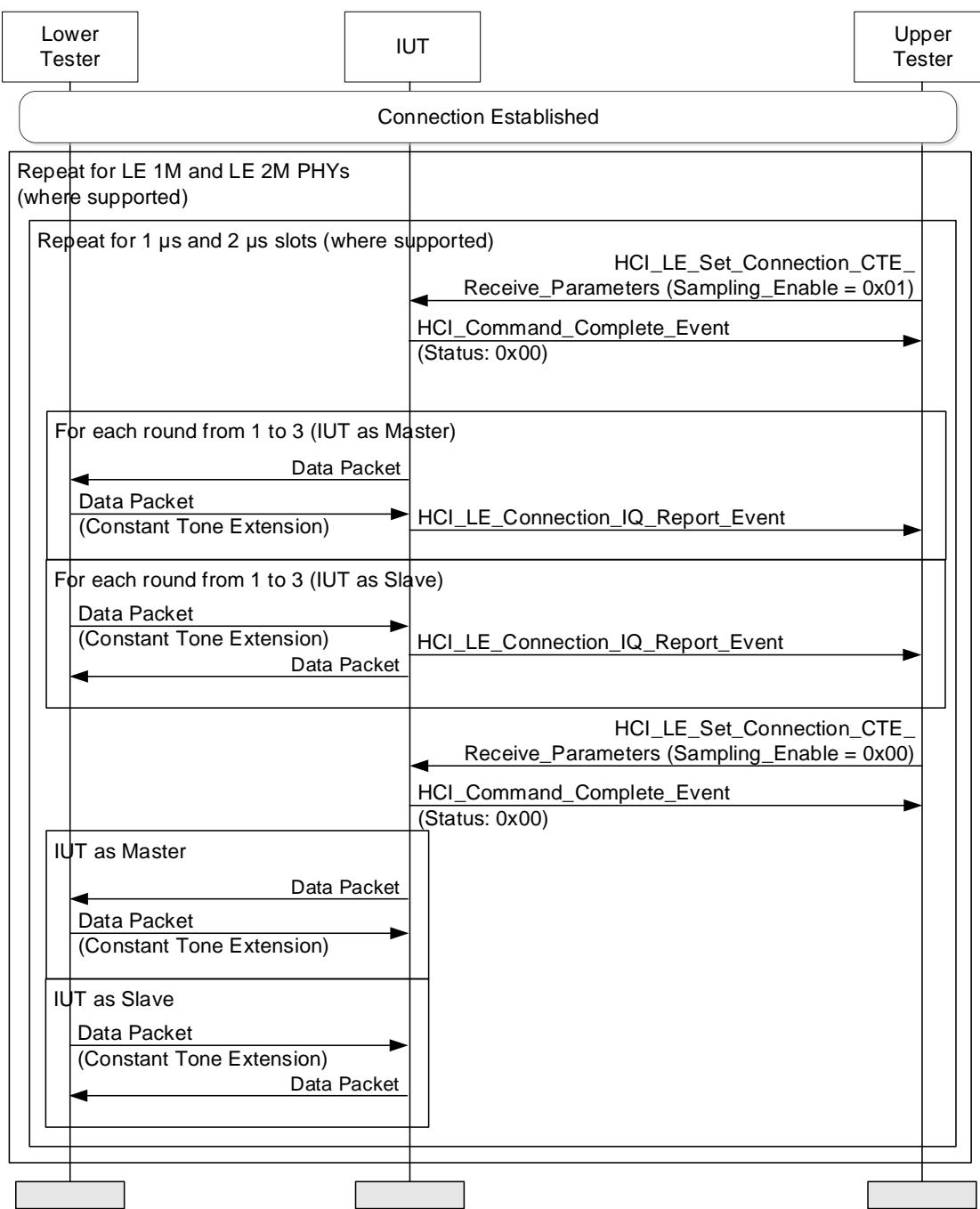


Figure 4.293: [Unrequested Constant Tone Extension, IUT Receiving, AoD]

For each round from 1 to 3 based on Table 4.42:

Round	CTETime (Step 1)
1	0x02
2	0x0A



Round	CTETime (Step 1)
3	0x14

Table 4.55: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x01 (enable).
2. The Lower Tester sends a data channel PDU to the IUT. The data channel PDU contains the CTEInfo field, with CTETime set to the value specified in [Table 4.55](#), RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 µs slots). The packet containing the data channel PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the specified CTETime.
3. The Upper Tester expects an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the data channel PDU was received, and RSSI shall be set to a valid value. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots). The connEventCounter shall be set to the connection event counter value of the data packet PDU.
4. Repeat steps 2-3 for each round shown in [Table 4.55](#).
5. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x00 (disable).
6. Repeat step 2.
7. The Upper Tester does not expect an HCI_LE_Connection_IQ_Report event from the IUT.
8. The Lower Tester expects the IUT to maintain the connection.
9. If the IUT supports 1 µs slots, repeat steps 1-4, except that in step 2 CTEType is set to 1 (AoD Constant Tone Extension with 1 µs slots) and that in step 3 CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 µs slots).
10. If the IUT supports the LE 2M PHY, switch to the LE 2M PHY via the PHY Update Procedure or via disconnecting and reconnecting on the LE 2M PHY and repeat steps 1-5.

- Expected Outcome

Pass Verdict

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

- The IUT generates an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x01.
- The IUT does not generate an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x00.
- The IUT acknowledges the Data Channel PDUs containing a Constant Tone Extension sent by the Lower Tester.

[4.3.6.17 \[Constant Tone Extension Request Procedure, IUT Initiated, AoA\]](#)

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.



The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Test Case IDs

LL/CON/SLA/BV-60-C

LL/CON/MAS/BV-56-C

- Reference

[10] 2.5, 5.1.12

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The antenna count is defined by the TSPX number of antennae IXIT entry.

- Test Procedure

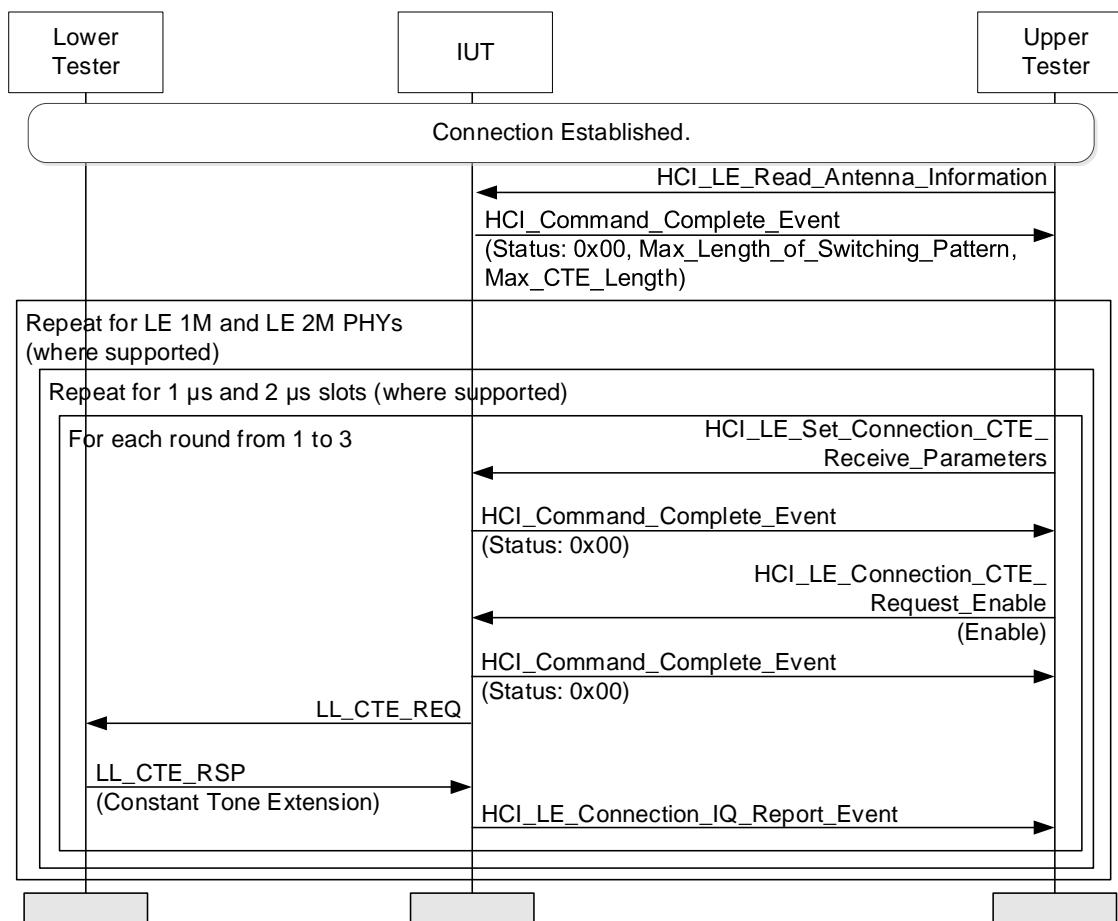


Figure 4.294: [Constant Tone Extension Request Procedure, IUT Initiated, AoA]

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_Length_Switching_Pattern between 0x02 and 0x4B. The Upper Tester stores the Max_Length_Switching_Pattern for future use.

For each round from 1 to 3 based on [Table 4.56](#):

Round	Requested_CTE_Length (Step 3, 5)
1	0x02
2	0x0A
3	0x14

Table 4.56: Parameter values for each case variation

2. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
 3. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to the value specified in [Table 4.56](#). Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension).
 4. The Lower Tester expects an LL_CTE_REQ PDU. MinCTELenReq shall be set to the Requested_CTE_Length value from step 3. CTETypeReq shall be set to 0 (AoA Constant Tone Extension).
 5. The Lower Tester sends an LL_CTE_RSP PDU. The LL_CTE_RSP PDU contains the CTEInfo field, with CTETime set to the MinCTELenReq value from step 4, RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in [Table 4.56](#).
 6. The Upper Tester expects an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the LL_CTE_RSP PDU was received, and RSSI shall be set to a valid value. RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array from Step 2. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). Slot_Durations shall be set to 0x02 (2 µs slots). The connEventCounter shall be set to the connection event counter value of the LL_CTE_RSP PDU. Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 4$. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.
 7. Repeat steps 2-6 for each round shown in [Table 4.56](#).
 8. If the IUT supports 1 µs slots, repeat steps 2-7, except that in step 2 Slot_Durations shall be set to 0x01 (1 µs slots) and that in step 6 Slot_Durations shall be set to 0x01 (1 µs slots) and Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 2$.
 9. If the IUT supports the LE 2M PHY, switch to the LE 2M PHY via the PHY Update Procedure or via disconnecting and reconnecting on the LE 2M PHY and repeat steps 2-8.
- Expected Outcome

Pass Verdict

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



- The IUT generates an HCI_LE_Connection_IQ_Report event.

4.3.6.18 [Constant Tone Extension Request Procedure, IUT Initiated, Periodic]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when periodic requests are selected. Test that the IUT generates multiple CTE Requests at the correct interval and returns appropriate responses to the Upper Tester based on the Lower Tester responses.

The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates periodic CTE Requests, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-61-C](#)

[LL/CON/MAS/BV-57-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).



- Test Procedure

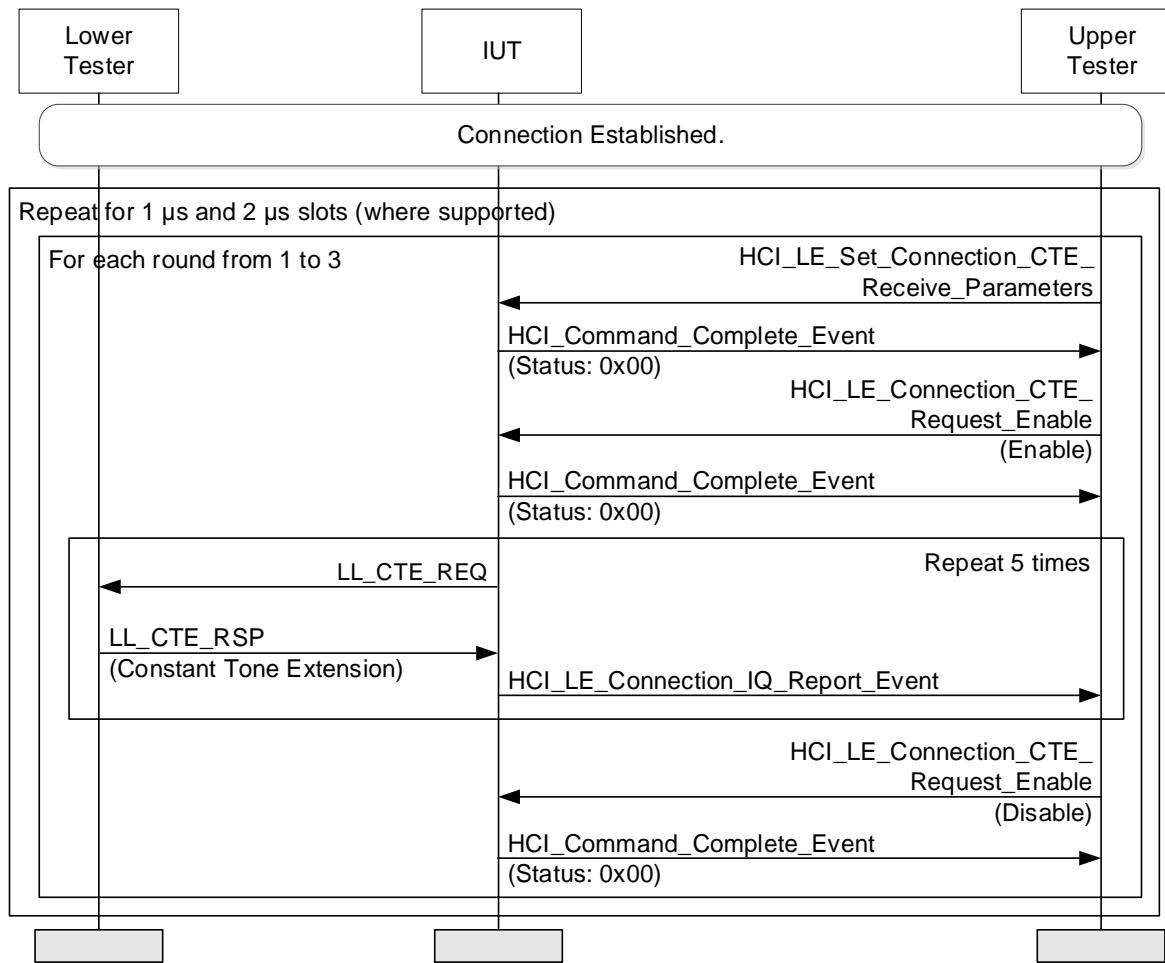


Figure 4.295: [Constant Tone Extension Request Procedure, IUT Initiated, Periodic]

For each round from 1 to 3 based on [Table 4.57](#):

Round	CTE_Request_Interval (Step 2, 4)
1	0x0001
2	0x0010
3	0x0167
4	A randomly selected value between 0x0080 (ca 5 s) and 0x0200 (ca 20 s) inclusive

Table 4.57: Parameter values for each case variation

- The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
- The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to the value specified in [Table 4.57](#).



Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots).

3. The Lower Tester expects an LL_CTE_REQ PDU.
4. The Lower Tester sends an LL_CTE_RSP PDU. The packet containing an LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in step 2.
5. The Upper Tester expects an HCI_LE_Connection_IQ_Report event from the IUT.
6. Repeat steps 3-5 until 5 Constant Tone Extensions have been transmitted. The Lower Tester expects to receive each successive LL_CTE_RSP PDU after a number of connection events equal to or greater than the CTE_Request_Interval value in [Table 4.57](#) since receiving the last such PDU.
7. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x00 (Disable).
8. Repeat steps 1-7 for each round shown in [Table 4.57](#).
9. If the IUT supports 1 μ s slots, repeat steps 1-8, except that in step 1 Slot_Durations shall be set to 0x01 (1 μ s slots) and that in step 5 Slot_Durations shall be set to 0x01 (1 μ s slots).

- Expected Outcome

Pass Verdict

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

- The IUT periodically requests Constant Tone Extensions, with at least CTE_Request_Interval connection events between requests.
- The IUT generates an HCI_LE_Connection_IQ_Report event.

[4.3.6.19 \[Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled\]](#)

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT correctly handles the case where the remote Lower Tester has Connection CTE Responses disabled.

The Lower Tester maintains a connection and has disabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and confirms the IUT correctly handles an LL_REJECT_EXT_IND PDU.

- Reference

[\[10\] 5.1.12](#)

- Test Case IDs

[LL/CON/SLA/BV-62-C](#)

[LL/CON/MAS/BV-58-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).



- Test Procedure

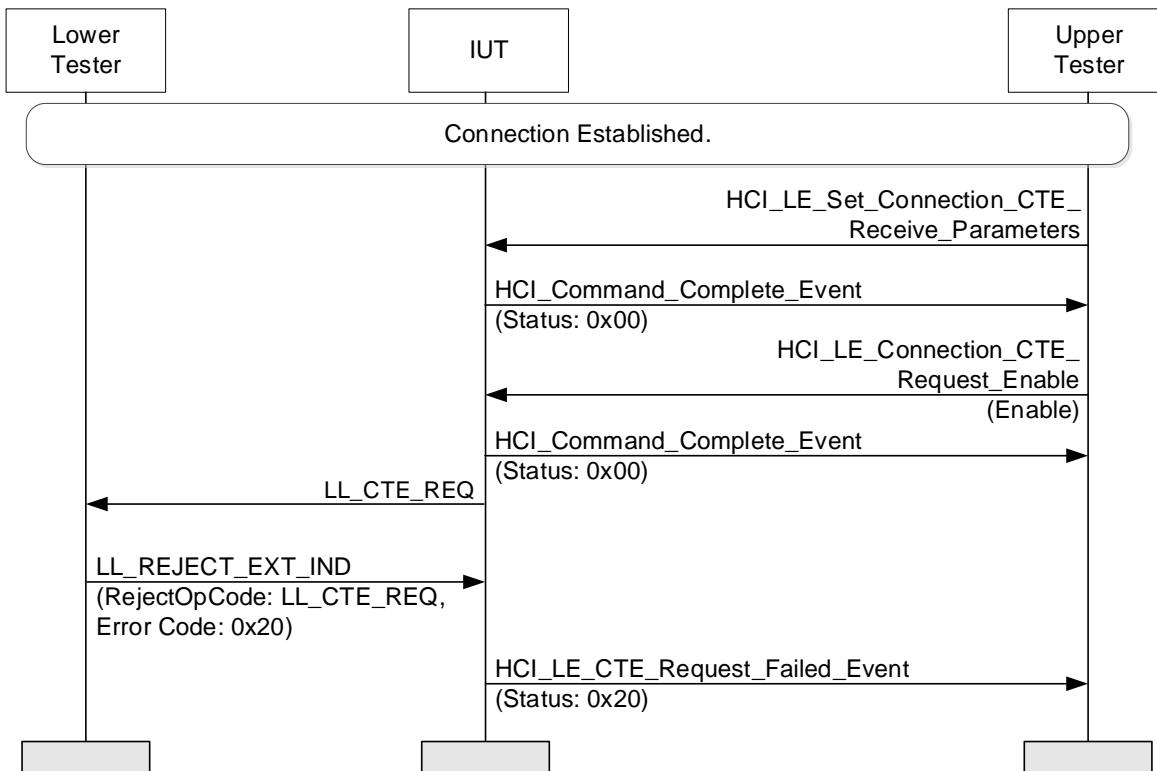


Figure 4.296: [Constant Tone Extension Request Procedure, IUT Initiated, Responses Disabled]

- The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02, Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
- The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 µs slots).
- The Lower Tester expects an LL_CTE_REQ PDU.
- The Lower Tester sends an LL_REJECT_EXT_IND PDU. RejectOpCode is set to LL_CTE_REQ. ErrorCode is set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).
- The Upper Tester expects an HCI_LE_CTE_Request_Failed event from the IUT. Status shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

- Expected Outcome

Pass Verdict

The IUT generates an HCI_LE_CTE_Request_Failed event with the Status set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).



4.3.6.20 [Constant Tone Extension Request Procedure, IUT Initiated, Timeout]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT correctly handles the case where the remote Lower Tester does not respond and the procedure times out.

The Lower Tester maintains a connection. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request. The Lower Tester ignores the request and allows the procedure to time out.

- Reference

[10] 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-64-C](#)

[LL/CON/MAS/BV-60-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

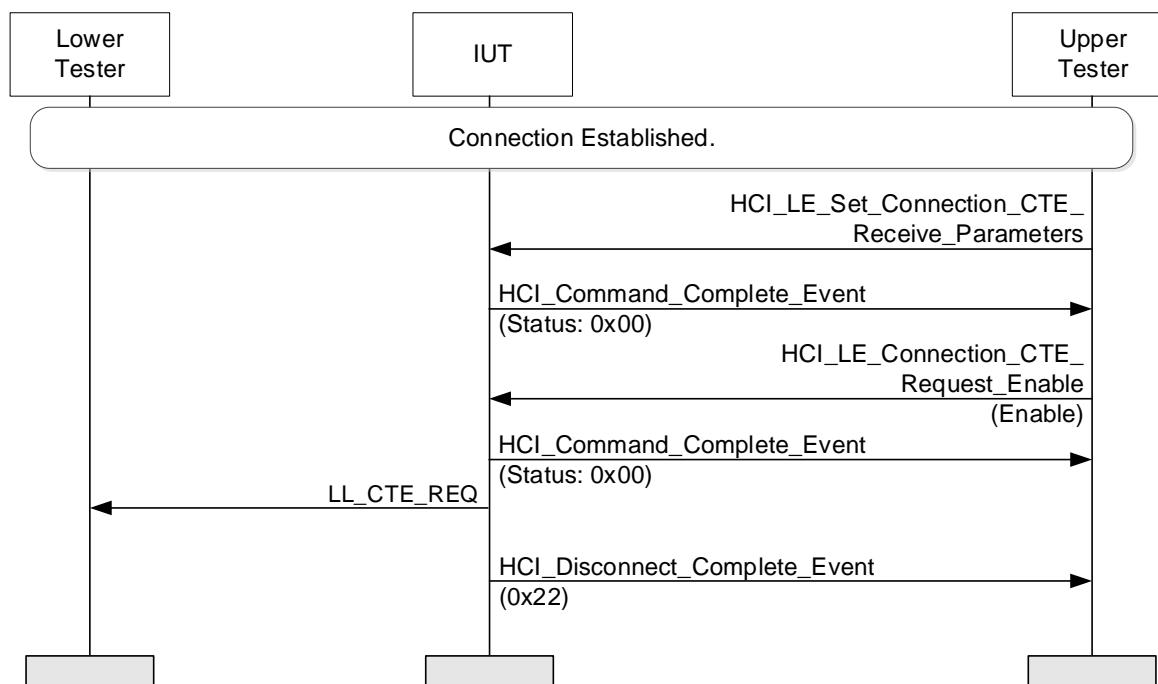


Figure 4.297: [Constant Tone Extension Request Procedure, IUT Initiated, Timeout]

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. If the IUT supports AoA Constant Tone Extension reception Length_of_Switching_Pattern shall be set to 0x02,



Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1, and Slot_Durations shall be set to 0x02 (2 μ s slots). Sampling_Enable shall be set to 0x01 (enable).

2. The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to 0x02. If the IUT supports AoA Constant Tone Extension reception, Requested_CTE_Type shall be set to 0x00 (AoA Constant Tone Extension); otherwise Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots).
3. The Lower Tester expects an LL_CTE_REQ PDU.
4. The Lower Tester does not respond, allowing the procedure to time out.
5. The Upper Tester expects an HCI_Disconnect_Complete event from the IUT with reason code set to 0x22 (LL Response Timeout).
6. The Lower Tester expects the IUT to stop maintaining the connection.

- Expected Outcome

Pass Verdict

The IUT disconnects the link after a procedure response timeout.

4.3.6.21 [Constant Tone Extension Request Procedure, IUT Responding, AoA]

- Test Purpose

Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains a connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT's response for proper formatting.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

LL/CON/SLA/BV-65-C

LL/CON/MAS/BV-61-C

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).



- Test Procedure

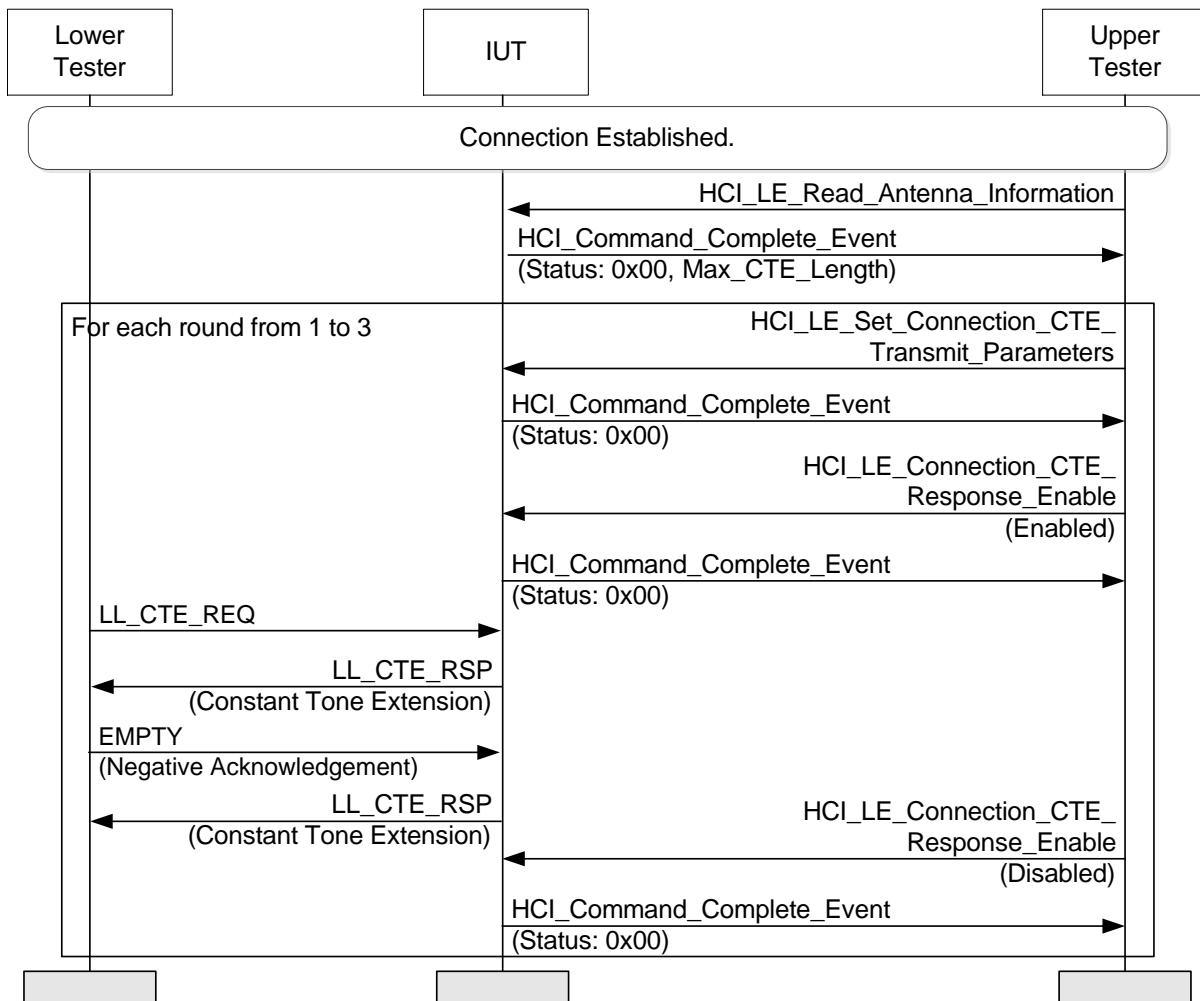


Figure 4.298: [Constant Tone Extension Request Procedure, IUT Responding, AoA]

1. The Upper Tester sends an HCI_LE_Read_Antenna_Information command to the IUT and expects the IUT to return a Max_CTE_Length between 0x02 and 0x14. The Upper Tester stores the Max_CTE_Length for future use.

For each round from 1 to 3 based on [Table 4.58](#):

Round	Requested_CTE_Length (Step 2, 5-6)
1	0x02
2	0x0A
3	Max_CTE_Length

Table 4.58: Parameter values for each case variation

2. If the CTETime listed in [Table 4.58](#) for this round is less than or equal to the Max_CTE_Length proceed to step 3; otherwise skip to step 8.
3. The Upper Tester sends an HCI_Command_Set_Connection_CTE_Transmit_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. CTE_Types shall be set to Allow AoA CTE Response (00000001b).



4. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).
 5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to the Requested_CTE_Length value specified in [Table 4.58](#). CTETypeReq is set to 0 (AoA Constant Tone Extension).
 6. The Lower Tester expects an LL_CTE_RSP PDU. The LL_CTE_RSP PDU shall contain the CTEInfo field, with CTETime set to a value greater than or equal to the MinCTELenReq value from step 5, RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing an LL_CTE_RSP PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime.
 7. The Lower Tester sends an Empty PDU to the IUT not acknowledging the LL_CTE_RSP PDU.
 8. The Lower Tester expects to receive an additional LL_CTE_RSP PDU with the same fields as in step 6.
 9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).
 10. Repeat steps 1-9 for each round shown in [Table 4.58](#).
- Expected Outcome

Pass Verdict

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

- The Lower Tester receives the CTE Response.

[4.3.6.22 \[Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled\]](#)

- Test Purpose

Tests that an IUT responds correctly to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses have been disabled. Test that the IUT generates an LL_REJECT_EXT_IND PDU.

The Lower Tester maintains a connection. The Upper Tester configures the IUT to disable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and confirms the IUT correctly sends an LL_REJECT_EXT_IND PDU.

- Reference

[\[10\] 5.1.12](#)

- Test Case IDs

[LL/CON/SLA/BV-66-C](#)

[LL/CON/MAS/BV-62-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The IUT does not have CTE Responses enabled for the connection.



- Test Procedure

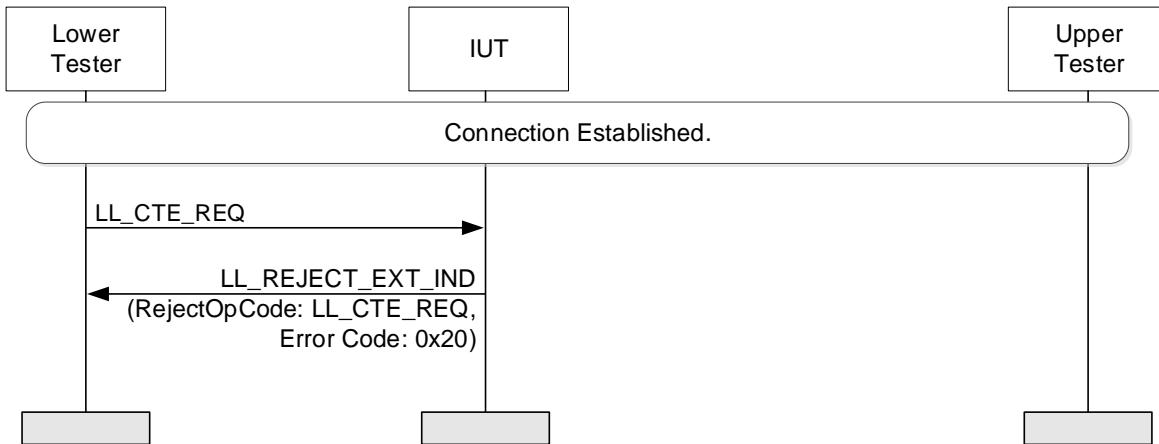


Figure 4.299: [Constant Tone Extension Request Procedure, IUT Responding, Responses Disabled]

1. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to the 0x02. If the IUT supports AoA CTE Transmission CTETypeReq is set to 0 (AoA Constant Tone Extension); otherwise CTETypeReq is set to 2 (AoD Constant Tone Extension with 2 us slots).
 2. The Lower Tester expects an LL_REJECT_EXT_IND PDU. RejectOpCode shall be set to LL_CTE_REQ. ErrorCode shall be set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).
- Expected Outcome

Pass Verdict

The IUT sends an LL_REJECT_EXT_IND PDU with RejectOpCode set to LL_CTE_REQ and ErrorCode set to 0x20 (Unsupported LMP Parameter Value/Unsupported LL Parameter Value).

4.3.6.23 [Constant Tone Extension Request Procedure, IUT Initiated, AoD]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains a connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

LL/CON/SLA/BV-67-C

LL/CON/MAS/BV-63-C

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).



- Test Procedure

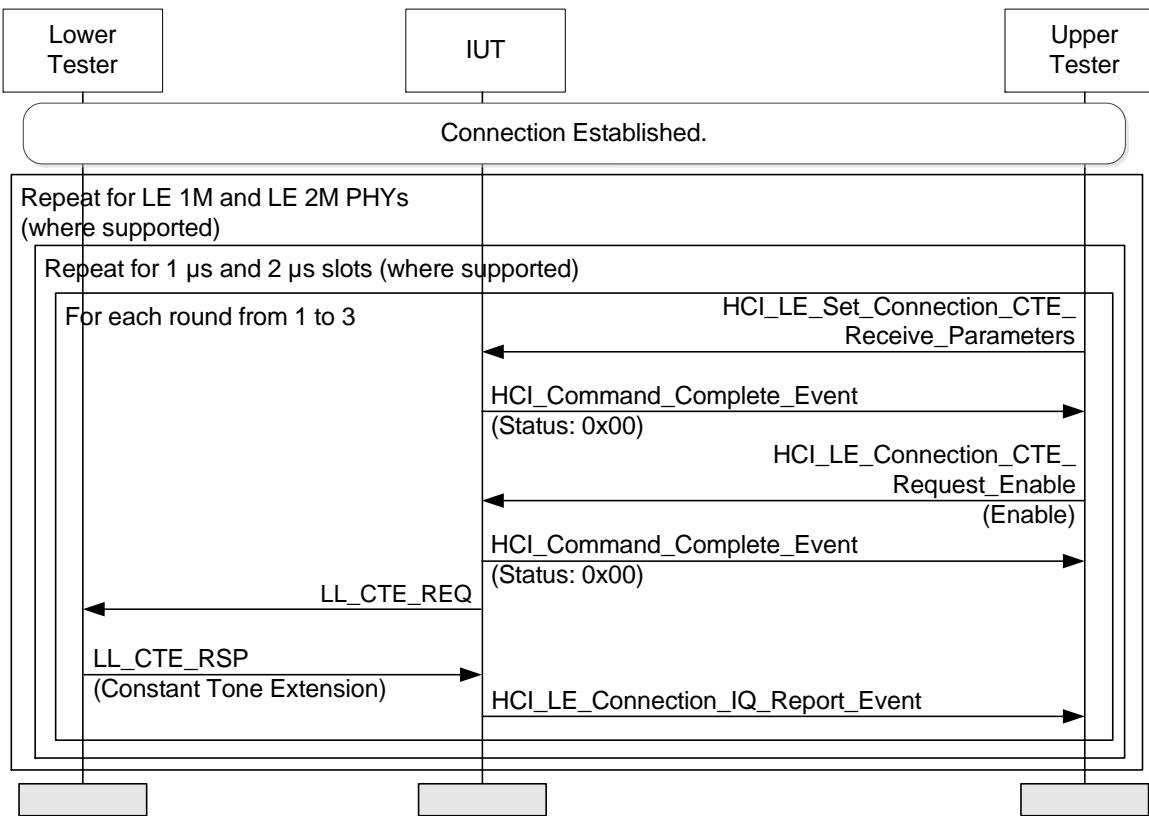


Figure 4.300: [Constant Tone Extension Request Procedure, IUT Initiated, AoD]

For each round from 1 to 3 based on [Table 4.59](#):

Round	Requested_CTE_Length (Step 2, 4)
1	0x02
2	0x0A
3	0x14

Table 4.59: Parameter values for each case variation

- The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x01 (enable).
- The Upper Tester sends an HCI_LE_Connection_CTE_Request_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set 0x01 (Enable). CTE_Request_Interval shall be set to 0x0000 (send LL_CTE_REQ at once). Requested_CTE_Length shall be set to the value specified in [Table 4.59](#). Requested_CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μs slots).
- The Lower Tester expects an LL_CTE_REQ PDU. MinCTELenReq shall be set to the Requested_CTE_Length value from step 2. CTETypeReq shall be set to 2 (AoD Constant Tone Extension with 2 μs slots).
- The Lower Tester sends an LL_CTE_RSP PDU. The LL_CTE_RSP PDU contains the CTEInfo field, with CTETime set to the MinCTELenReq value from step 3, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 μs slots). The packet containing an



LL_CTE_RSP PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the Requested_CTE_Length specified in [Table 4.59](#). While transmitting the Constant Tone Extension field, the Lower Tester switches antennae using 2 μ s slots.

5. The Upper Tester expects an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the LL_CTE_RSP PDU was received, and RSSI shall be set to a valid value. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x02 (AoD Constant Tone Extension with 2 μ s slots). The connEventCounter shall be set to the connection event counter value of the LL_CTE_RSP PDU. Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 4$. I_Sample[0] through I_Sample[Sample_Count - 1] and Q_Sample[0] through Q_Sample[Sample_Count - 1] shall each be set to a signed integer.
6. Repeat steps 1-5 for each round shown in [Table 4.59](#).
7. If the IUT supports 1 μ s slots, repeat steps 1-6, except that in step 2 Requested_CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 μ s slots), that in step 3 CTETypeReq shall be set to 1 (AoD Constant Tone Extension with 1 μ s slots), that in step 4 CTEType is set to 1 (AoD Constant Tone Extension with 1 μ s slots) and the Lower Tester switches antennae using 1 μ s slots, and that in step 5 the CTE_Type shall be set to 0x01 (AoD Constant Tone Extension with 1 μ s slots) and Sample_Count shall be set to $8 + (\text{CTETime} - 12) / 2$.

- Expected Outcome

Pass Verdict

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

- The IUT generates an HCI_LE_Connection_IQ_Report event.

4.3.6.24 [Constant Tone Extension Request Procedure, IUT Responding, AoD]

- Test Purpose

Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains a connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT's response for proper formatting.

- Reference

[\[10\]](#) 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-68-C](#)

[LL/CON/MAS/BV-64-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.



- Test Procedure

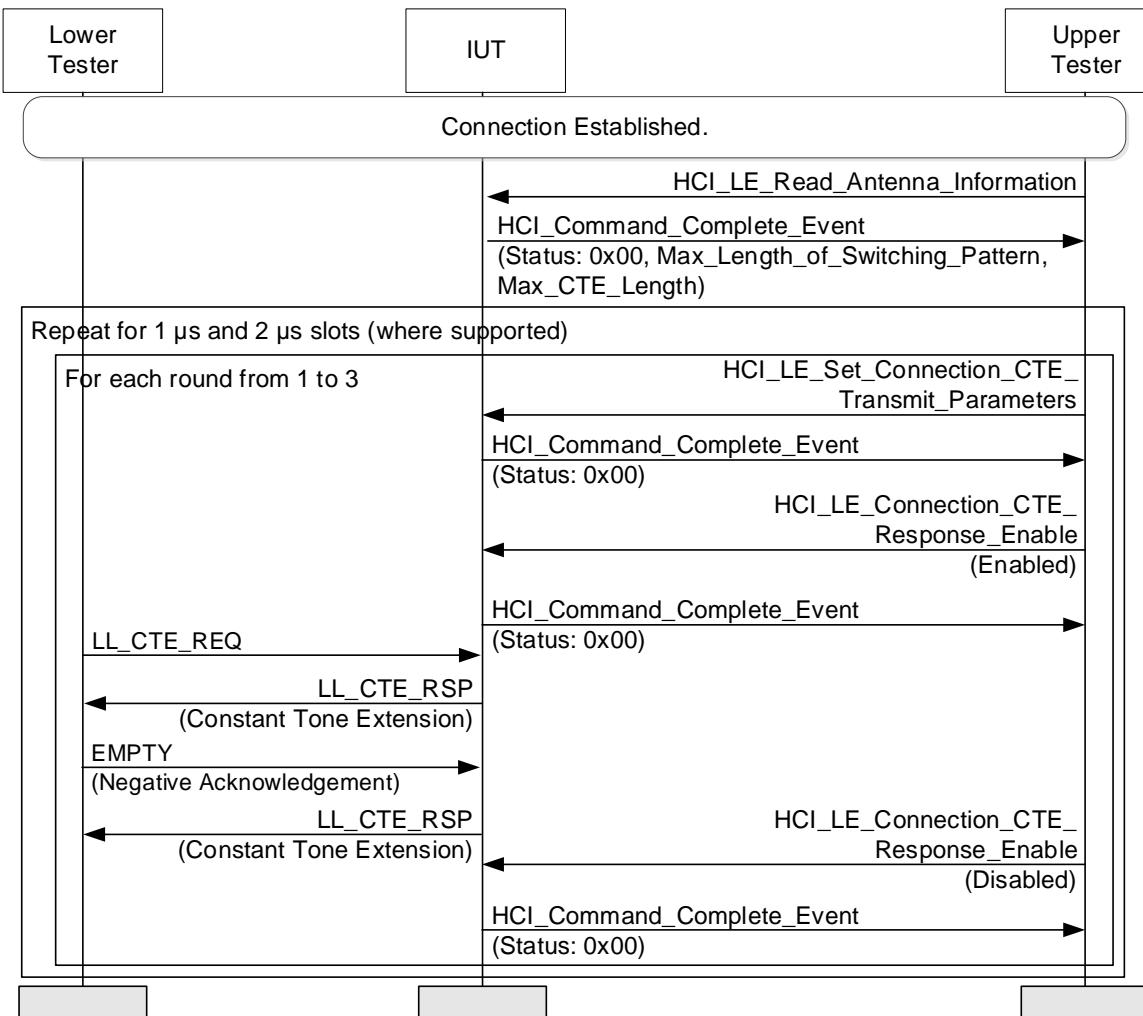


Figure 4.301: [Constant Tone Extension Request Procedure, IUT Responding, AoD]

1. The Upper Tester sends an **HCI_LE_Read_Antenna_Information** command to the IUT and expects the IUT to return a **Max_Length_Switching_Pattern** between 0x02 and 0x4B and a **Max_CTE_Length** between 0x02 and 0x14. The Upper Tester stores the **Max_Length_Switching_Pattern** and the **Max_CTE_Length** for future use.

For each round from 1 to 3 based on [Table 4.60](#):

Round	Requested_CTE_Length (Step 2, 5-6)
1	0x02
2	0x0A
3	Max_CTE_Length

Table 4.60: Parameter values for each case variation

2. If the CTETime listed in [Table 4.60](#) for this round is less than or equal to the **Max_CTE_Length** proceed to step 3; otherwise skip to step 8.
3. The Upper Tester sends an **HCI_LE_Set_Connection_CTE_Transmit_Parameters** command to the IUT. Connection_Handle shall be set to the handle of the connection.



Length_of_Switching_Pattern shall be set to Max_Length_Switching_Pattern. Antenna_IDs[0] through Antenna_IDs[Length_of_Switching_Pattern - 1] shall be set to the pattern 0, 1, ..., TSPX_number_of_antennae, with the pattern repeated and truncated as necessary to specify Antenna_IDs[] values. CTE_Types shall be set to Allow AoD CTE Response with 2 μ s slots (00000100b).

4. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Enabled (0x01).
 5. The Lower Tester sends an LL_CTE_REQ PDU. MinCTELenReq is set to the Requested_CTE_Length value specified in [Table 4.60](#). CTETypeReq is set to 2 (AoD Constant Tone Extension with 2 μ s slots).
 6. The Lower Tester expects an LL_CTE_RSP PDU. The LL_CTE_RSP PDU shall contain the CTEInfo field, with CTETime set to a value greater than or equal to the MinCTELenReq value from step 5, RFU set to '0', and the CTEType set to 2 (AoD Constant Tone Extension with 2 μ s slots). The packet containing an LL_CTE_RSP PDU shall contain the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the CTETime.
 7. The Lower Tester sends an Empty PDU to the IUT not acknowledging the LL_CTE_RSP PDU.
 8. The Lower Tester expects to receive an additional LL_CTE_RSP PDU with the same fields as in step 6.
 9. The Upper Tester sends an HCI_LE_Connection_CTE_Response_Enable command to the IUT. Connection_Handle shall be set to the handle of the connection. Enable shall be set to Disabled (0x00).
 10. Repeat steps 1-9 for each round shown in [Table 4.60](#).
 11. If the IUT supports 1 μ s slots, repeat steps 1-10, except that in step 3 CTE_Types shall be set to Allow AoD CTE Response with 1 μ s slots (00000010b), that in step 5 the CTETypeReq shall be set to 1 (AoD Constant Tone Extension with 1 μ s slots), and that in step 6 CTEType shall be set to 1 (AoD Constant Tone Extension with 1 μ s slots).
- Expected Outcome

Pass Verdict

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

- The Lower Tester receives the CTE Response.

[4.3.6.25 \[Unrequested Constant Tone Extension, IUT Receiving, AoA\]](#)

- Test Purpose

Tests that an IUT is process an unrequested AoA Constant Tone Extension from a remote Lower Tester. Test that the IUT sends the appropriate event to the Upper Tester based on the Lower Tester's Constant Tone Extension.

The Lower Tester maintains a connection and transmits an AoA Constant Tone Extension on a data packet. The Upper Tester observes the event returned by the IUT.

- Reference

[\[10\]](#) 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-69-C](#)

[LL/CON/MAS/BV-65-C](#)



- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The antenna count is defined by the TSPX_number_of_antennae IXIT entry.



- Test Procedure

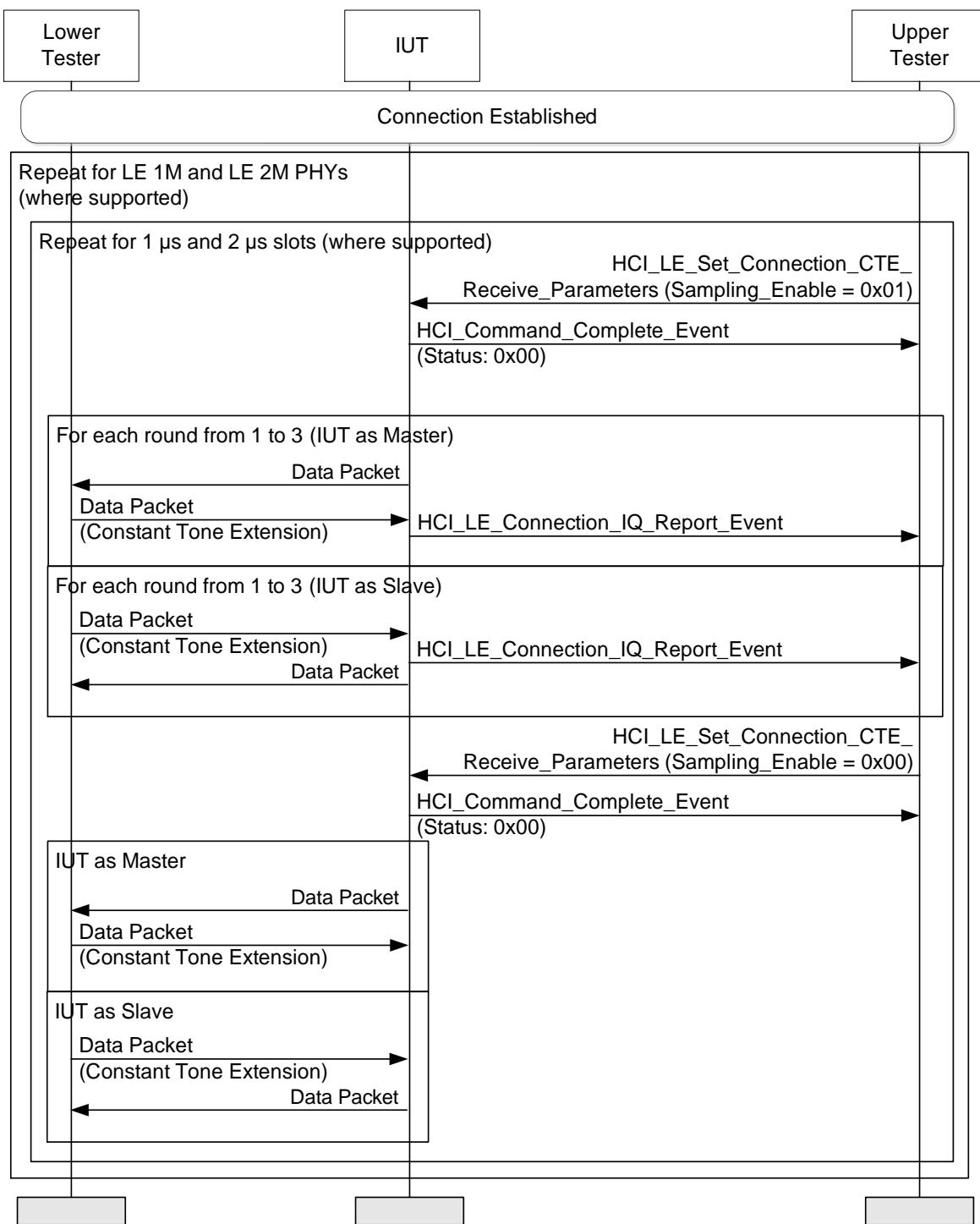


Figure 4.302: [Unrequested Constant Tone Extension, IUT Receiving, AoA]

For each round from 1 to 3 based on [Table 4.61](#):

Round	CTETime (Step 1)
1	0x02
2	0x0A
3	0x14

Table 4.61: Parameter values for each case variation

1. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Length_of_Switching_Pattern shall be set to 0x02. Antenna_IDs[0] and Antenna_IDs[1] shall be set to the pattern 0, 1. Slot_Durations shall be set to 0x02 (2 µs slots). Sampling_Enable shall be set to 0x01 (enable).
 2. The Lower Tester sends a data channel PDU to the IUT. The data channel PDU contains the CTETime field, with CTETime set to the value specified in [Table 4.61](#), RFU set to '0', and the CTEType set to 0 (AoA Constant Tone Extension). The packet containing the data channel PDU contains the Constant Tone Extension field, a constantly modulated series of unwhitened 1s, following the CRC of length matching the specified CTETime.
 3. The Upper Tester expects an HCI_LE_Connection_IQ_Report event from the IUT. Connection_Handle shall be set to the handle of the connection. RX_PHY and Data_Channel_Index shall be set to the correct values indicating how the data channel PDU was received, and RSSI shall be set to a valid value. RSSI_Antenna_ID shall be set to a value from the Antenna_IDs array from Step 1. The Packet_Status shall be set to 0x00 (CRC was correct). CTE_Type shall be set to 0x00 (AoA Constant Tone Extension). The connEventCounter shall be set to the connection event counter value of the data packet PDU.
 4. Repeat steps 2-3 for each round shown in [Table 4.61](#).
 5. The Upper Tester sends an HCI_LE_Set_Connection_CTE_Receive_Parameters command to the IUT. Connection_Handle shall be set to the handle of the connection. Sampling_Enable shall be set to 0x00 (disable).
 6. Repeat step 2.
 7. The Lower Tester expects the IUT to maintain the connection.
 8. The Upper Tester does not expect an HCI_LE_Connection_IQ_Report event from the IUT.
 9. If the IUT supports the LE 2M PHY, switch to the LE 2M PHY via the PHY Update Procedure or via disconnecting and reconnecting on the LE 2M PHY and repeat steps 2-4.
- Expected Outcome

Pass Verdict

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

- The IUT generates an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x01.
- The IUT does not generate an HCI_LE_Connection_IQ_Report event when Sampling_Enable is set to 0x00.
- The IUT acknowledges the Data Channel PDUs containing a Constant Tone Extension sent by the Lower Tester.



4.3.6.26 [Constant Tone Extension Request Procedure, IUT Initiated, AoA, Encrypted Connection]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when the connection is encrypted. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains an encrypted connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-71-C](#)

[LL/CON/MAS/BV-67-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoA].

- Expected Outcome

Pass Verdict

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoA].

4.3.6.27 [Constant Tone Extension Request Procedure, IUT Responding, AoA, Encrypted Connection]

- Test Purpose

Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester on an encrypted connection when Connection CTE Responses are enabled. Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains an encrypted connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT's response for proper formatting.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-72-C](#)

[LL/CON/MAS/BV-68-C](#)



- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Responding, AoA].

- Expected Outcome

Pass Verdict

Same as [Constant Tone Extension Request Procedure, IUT Responding, AoA].

4.3.6.28 [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Encrypted Connection]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester when the connection is encrypted. Test that the IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains an encrypted connection and has enabled Connection CTE Responses. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-73-C](#)

[LL/CON/MAS/BV-69-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoD].

- Expected Outcome

Pass Verdict

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoD].

4.3.6.29 [Constant Tone Extension Request Procedure, IUT Responding, AoD, Encrypted Connection]

- Test Purpose

Tests that an IUT is able to respond to a Constant Tone Extension Request Procedure initiated by a remote Lower Tester on an encrypted connection when Connection CTE Responses are enabled.



Test that the IUT generates a LL_CTE_RSP PDU with the correct Constant Tone Extension field format.

The Lower Tester maintains an encrypted connection. The Upper Tester configures the IUT to enable Connection CTE Responses. Lower Tester initiates a single CTE Request to the IUT and observes the IUT's response for proper formatting.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-74-C](#)

[LL/CON/MAS/BV-70-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The IUT's antenna count is defined by the TSPX_number_of_antennae IXIT entry.

- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Responding, AoD].

- Expected Outcome

Pass Verdict

Same as [Constant Tone Extension Request Procedure, IUT Responding, AoD].

4.3.6.30 [Constant Tone Extension Request Procedure, IUT Initiated, AoA, Incorrect CRC]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoA CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains a connection and has been configured to transmit Connection CTE Responses with incorrect CRCs. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-75-C](#)

[LL/CON/MAS/BV-71-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

The antenna count is defined by the TSPX_number_of_antennae IXIT entry.



- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoA], except that in step 5, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC, and that in step 6, the Packet_Status shall be set to 0x01 or 0x02.

- Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connection_IQ_Report events with Packet_Status set to indicate the CRC was incorrect.

Inconclusive Verdict

The IUT does not generate any HCI_LE_Connection_IQ_Report events.

4.3.6.31 [Constant Tone Extension Request Procedure, IUT Initiated, AoD, Incorrect CRC]

- Test Purpose

Tests that an IUT is able to initiate the Constant Tone Extension Request Procedure with a remote Lower Tester. Test that the IUT generates a single AoD CTE Request and returns appropriate response to the Upper Tester based on the Lower Tester response.

The Lower Tester maintains a connection and has been configured to transmit Connection CTE Responses with incorrect CRCs. The Upper Tester configures the IUT for CTE Requests, initiates a single CTE Request, and observes the responses returned by the IUT.

- Reference

[10] 2.5, 5.1.12

- Test Case IDs

[LL/CON/SLA/BV-76-C](#)

[LL/CON/MAS/BV-72-C](#)

- Initial Condition

MAS/SLA: Connected (LE 1M PHY, any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value).

- Test Procedure

Same as [Constant Tone Extension Request Procedure, IUT Initiated, AoA], except that in step 4, the Lower Tester transmits the packet containing the Constant Tone Extension field using an incorrect CRC, and that in step 5, the Packet_Status shall be set to 0x01 or 0x02.

- Expected Outcome

Pass Verdict

The IUT generates HCI_LE_Connection_IQ_Report events with Packet_Status set to indicate the CRC was incorrect.

Inconclusive Verdict

The IUT does not generate any HCI_LE_Connection_IQ_Report events.



4.4 TIM

Tests that the IUT behaves according to timing requirements in procedures where there may be variable timing. The timing requirements apply for event intervals when the IUT is in low power mode and for packet timing and intervals when the IUT is in active mode.

4.4.1 ADV

Tests that the IUT behaves according to timing constraints as an advertiser.

4.4.1.1 LL/TIM/ADV/BV-01-C [Earliest Transmission to Advertiser]

- Test Purpose

Test that an advertiser IUT responds to a scan request sent using the minimum timing between packets ($T_{IFS} - 2 \mu\text{sec}$).

The Lower Tester requests information from the IUT and expects a response.

- Reference

[3] 4.2, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

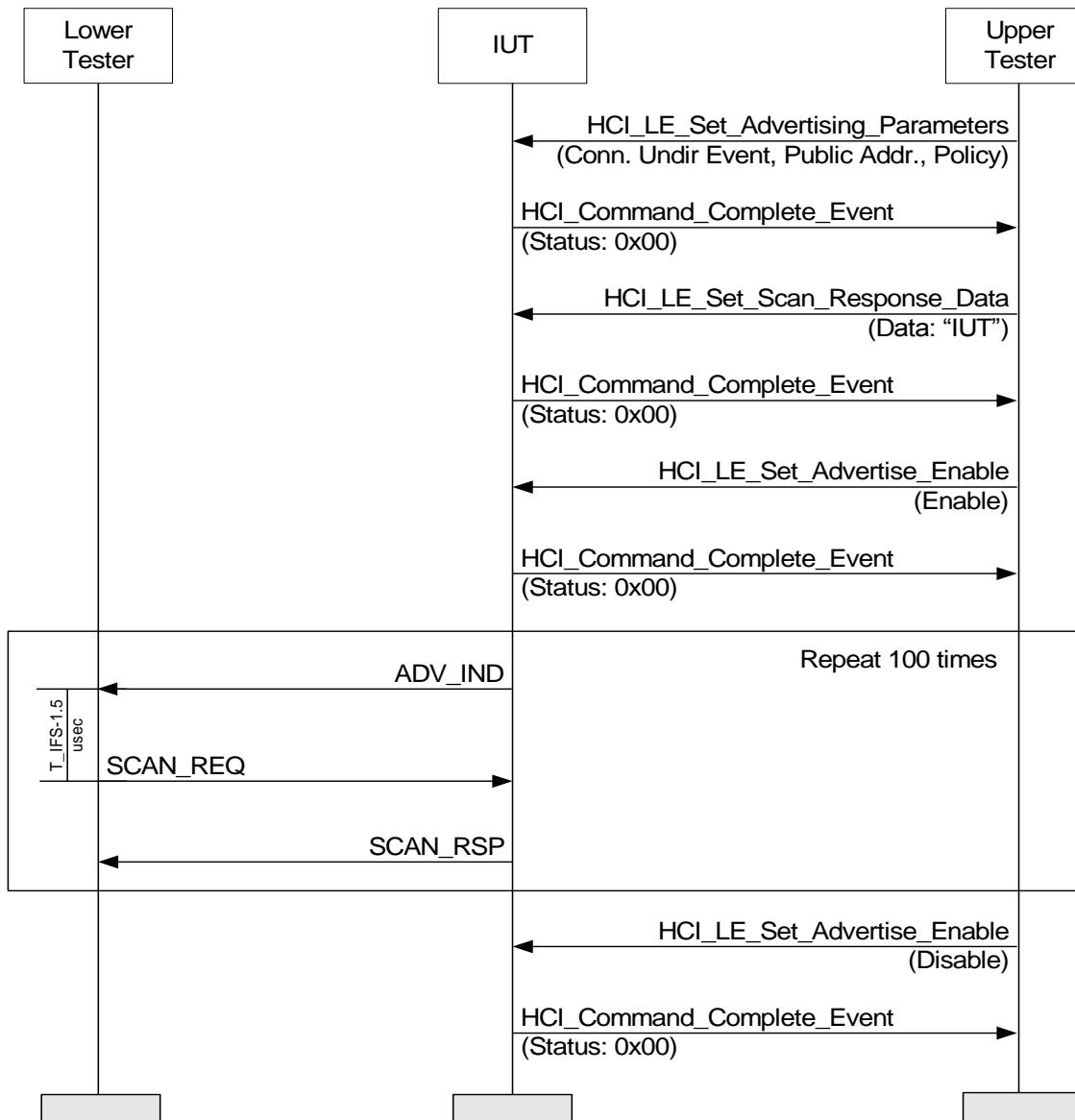


Figure 4.303: LL/TIM/ADV/BV-01-C [Earliest transmission to Advertiser]

- Configure Lower Tester to start active scanning.
 - Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
 - Lower Tester expects an ADV_IND packet from the IUT and responds with a SCAN_REQ packet, using the minimum time after the end of the advertising packet ($T_{IFS} - 1.5 \mu\text{sec}$).
 - Lower Tester expects a SCAN_RSP packet from the IUT T_{IFS} after the SCAN_REQ.
 - Repeat steps 3-4 100 times.
- Expected Outcome

Pass Verdict

The IUT responds at least to 95% of the SCAN_REQ packets sent by the Lower Tester in Step 3.



- Notes

Lower Tester is configured to use 1.5 μ sec instead of 2 μ sec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.1.2 LL/TIM/ADV/BV-02-C [Latest Transmission to Advertiser]

- Test Purpose

Test that an advertiser IUT responds to a scan request sent using the maximum timing between packets ($T_{IFS} + 2 \mu\text{sec}$).

The Lower Tester requests information from the IUT and expects a response.

- Reference

[3] 4.2, 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

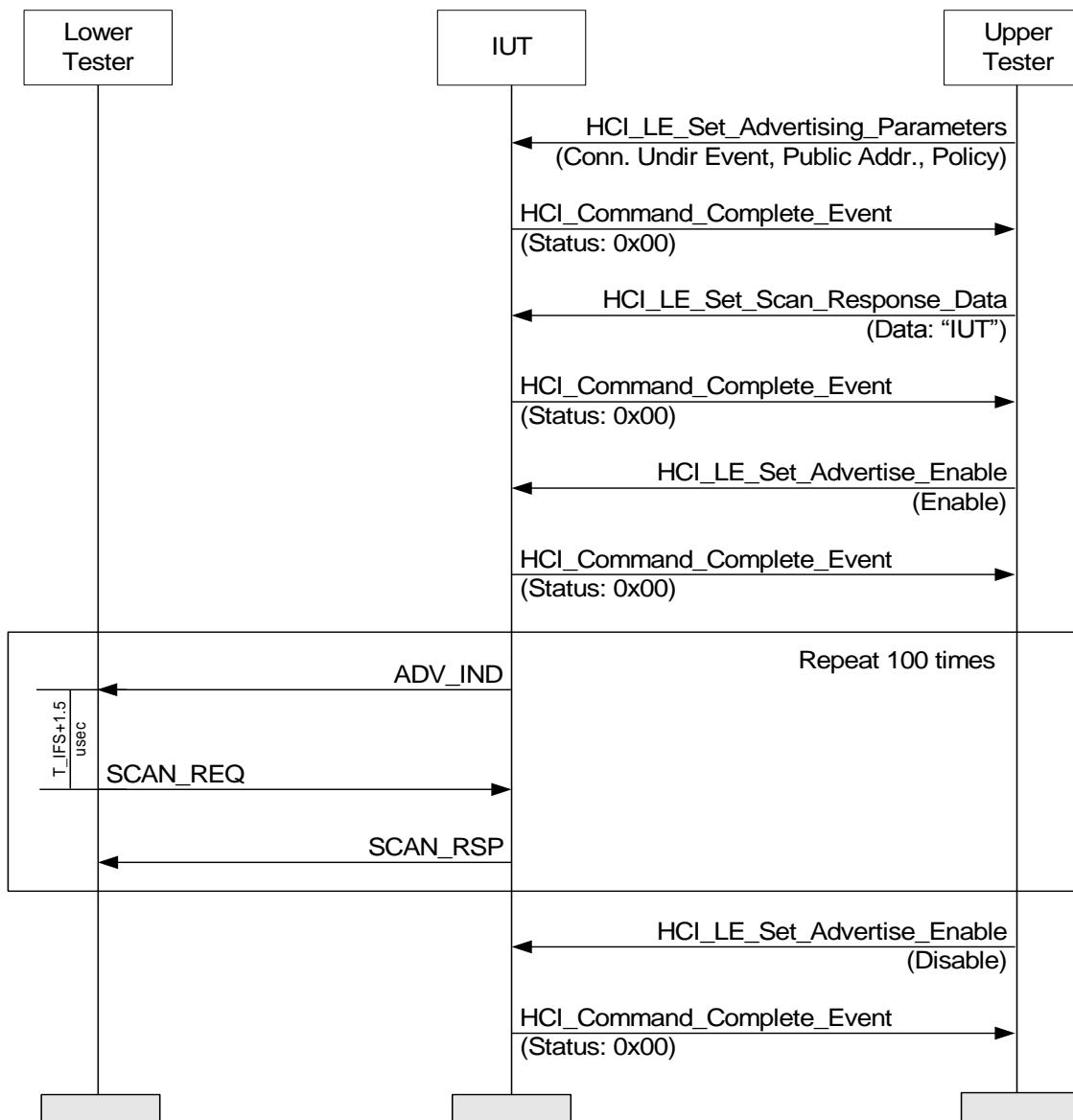


Figure 4.304: LL/TIM/ADV/BV-02-C [Latest transmission to Advertiser]

- Configure Lower Tester to start active scanning.
 - Upper Tester enables undirected advertising in the IUT using all supported advertising channels.
 - Lower Tester expects an `ADV_IND` packet from the IUT and responds with a `SCAN_REQ` packet, using the maximum time after the end of the advertising packet ($T_{IFS} + 1.5 \mu\text{sec}$).
 - Lower Tester expects a `SCAN_RSP` packet from the IUT T_{IFS} after the `SCAN_REQ`.
 - Repeat steps 3-4 100 times.
- Expected Outcome

Pass Verdict

The IUT responds at least to 95% of the `SCAN_REQ` packets sent by the Lower Tester in Step 3.



- Notes

Lower Tester is configured to use 1.5 μ sec instead of 2 μ sec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.1.3 LL/TIM/ADV/BV-03-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser]

- Test Purpose

Tests that an advertiser IUT sends scannable undirected ADV_EXT_IND with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel. Tests that an advertiser IUT responds to a scan request on the secondary channel when scanner uses the minimum transmission time to advertiser.

- Reference

[10] 4.4.2.5.4

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

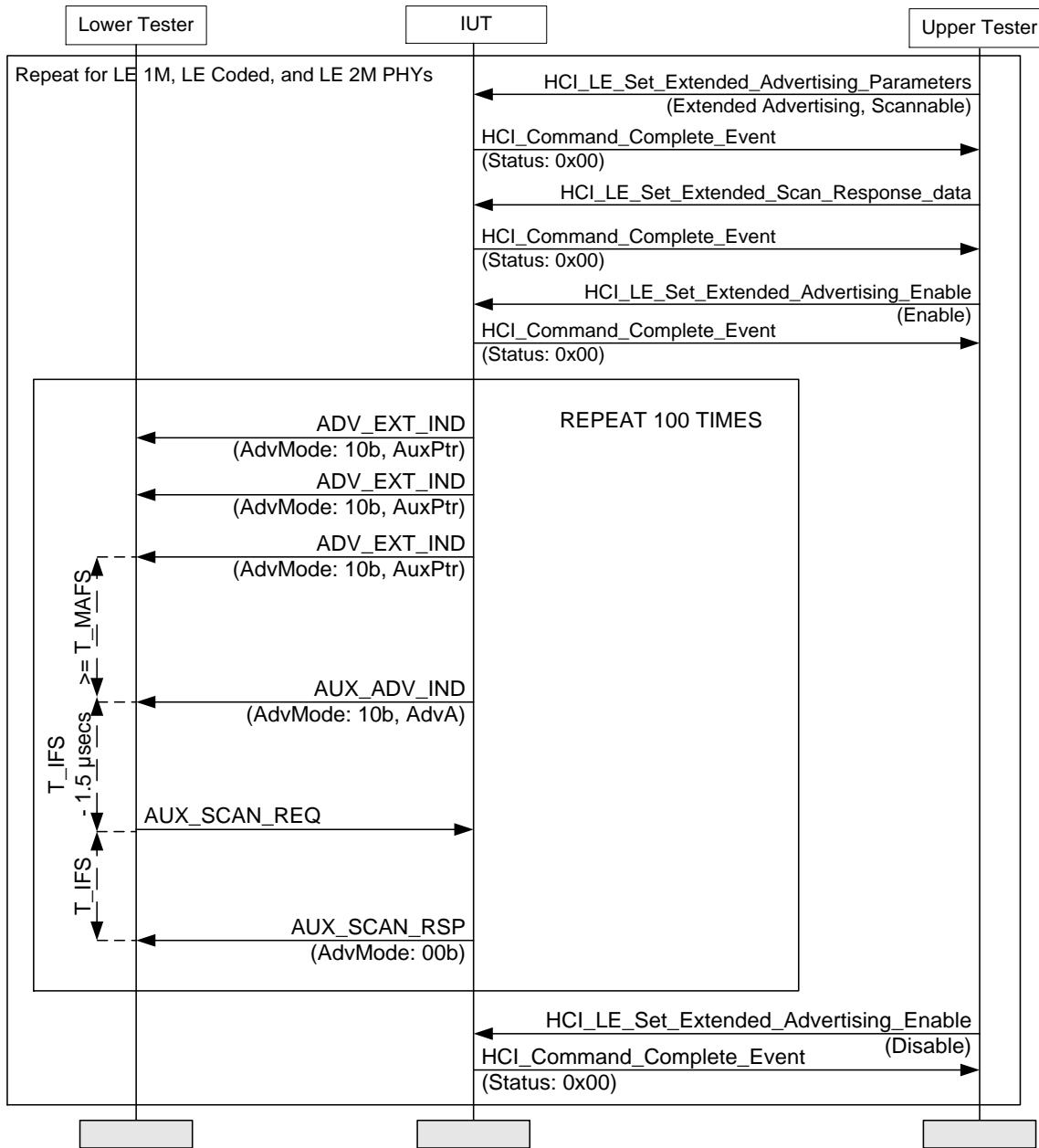


Figure 4.305: LL/TIM/ADV/BV-03-C [Extended Advertising, Secondary Channel, Earliest Transmission to Advertiser]

- The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter bit 1 (Scannable Advertising) shall be set and all other bits cleared. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
- The Upper Tester sends an `HCI_Command_Complete_Event` (Status: 0x00) to IUT.
- The Upper Tester sends an `HCI_Command_Complete_Event` (Enable) to IUT.
- The Upper Tester enables advertising using the `HCI_Command_Complete_Event` (Enable) command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

4. The Lower Tester expects an ADV_EXT_IND PDU with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent T_MAFS after the ADV_EXT_IND PDU or later. The Lower Tester sends an AUX_SCAN_REQ PDU the minimum time after the end of the AUX_ADV_IND PDU ($T_{IFS} - 1.5 \mu\text{sec}$).
 5. Lower Tester expects an AUX_SCAN_RSP packet from the IUT T_{IFS} (plus or minus 2 μsecs) after the AUX_SCAN_REQ.
 6. Repeat steps 4-5 100 times.
 7. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
 8. If the IUT supports the LE Coded PHY, repeat steps 1-7, except in Step 1, the Primary_Advertising_PHY shall be set to 0x03 (LE Coded), and the Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
 9. If the IUT supports the LE 2M PHY, repeat steps 1-7, except in Step 1, the Primary_Advertising_PHY shall be set to 0x01 (LE 1M), and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).
- Expected Outcome
- Pass Verdict
- The IUT responds to at least 95% of the AUX_SCAN_REQ packets sent by the Lower Tester in step 4 within T_{IFS} (plus or minus 2 μsecs).
- The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_{MAFS} .
- Notes
- Lower Tester is configured to use $1.5\mu\text{sec}$ instead of $2\mu\text{sec}$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.1.4 LL/TIM/ADV/BV-04-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser]

- Test Purpose
- Tests that an advertiser IUT sends scannable undirected ADV_EXT_IND PDUs with the AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel. Tests that an advertiser IUT responds to a scan request on the secondary channel when scanner uses the maximum transmission time to advertiser ($T_{IFS} + 2 \mu\text{sec}$).
- Reference
- [10] 4.4.2.5.4
- Initial Condition
- Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map
- State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

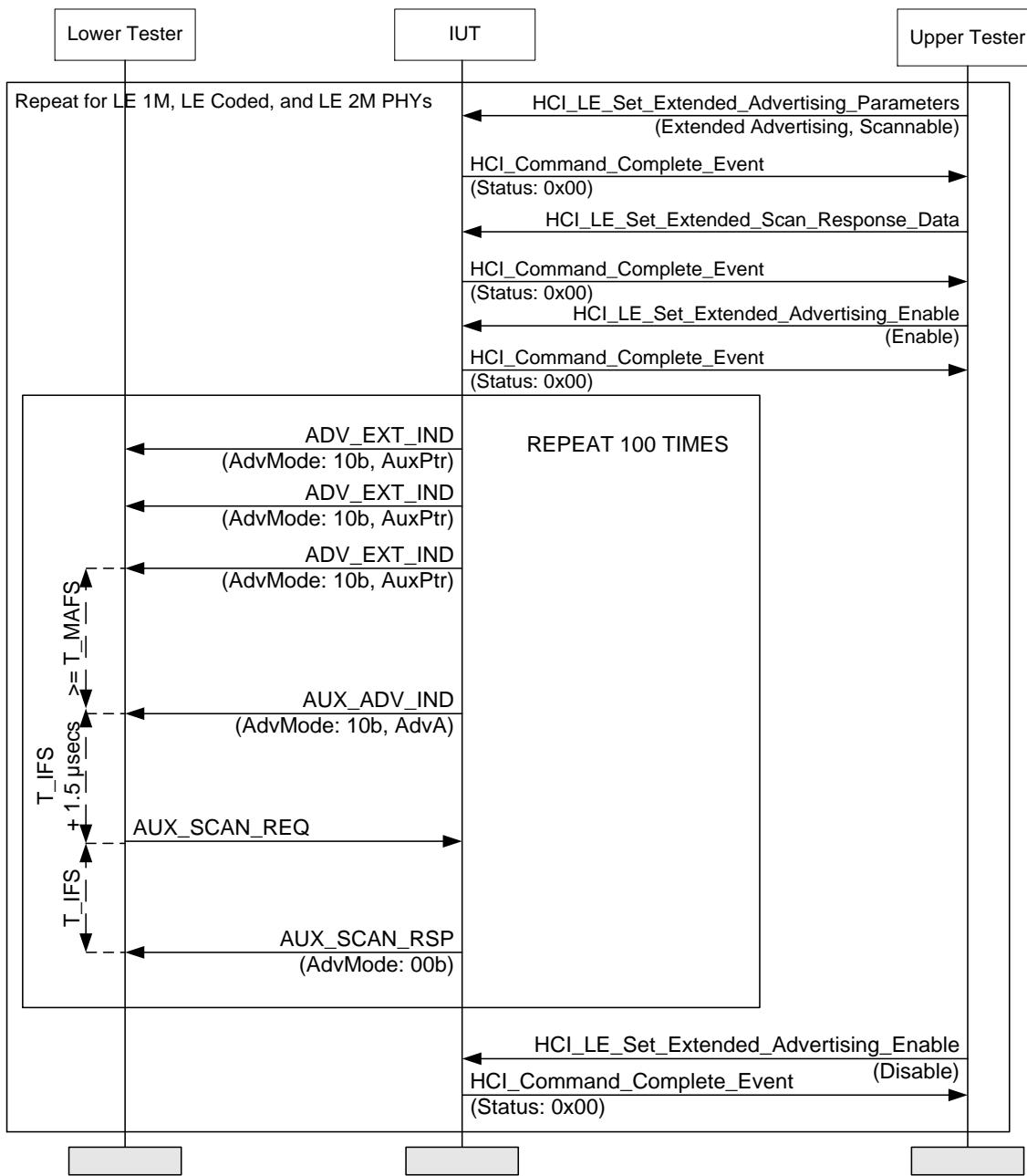


Figure 4.306: LL/TIM/ADV/BV-04-C [Extended Advertising, Secondary Channel, Latest Transmission to Advertiser]

- The Upper Tester sends an `HCI_LE_Set_Extended_Advertising_Parameters` command to the IUT using all supported advertising channels and minimum advertising interval. Advertising_Event_Properties parameter bit 1 (Scannable Advertising) shall be set and all other bits cleared. The Own_Address_Type shall be set to 0x00 (Public Device Address). The Primary_Advertising_PHY shall be set to 0x01 (LE 1M). The Secondary_Advertising_PHY shall be set to 0x01 (LE 1M).
- The Upper Tester sends an `HCI_LE_Set_Extended_Scan_Response_Data` command to the IUT containing at least 1 octet of Scan Response Data.
- The Upper Tester enables advertising using the `HCI_LE_Set_Extended_Advertising_Enable` command. The Duration[0] parameter is set to 0x0000 (No Advertising Duration).

4. Lower Tester receives an ADV_EXT_IND PDU with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent T_MAFS after the ADV_EXT_IND PDU or later. The Lower Tester sends an AUX_SCAN_REQ PDU the maximum time after the end of the AUX_ADV_IND PDU ($T_{IFS} + 1.5 \mu\text{sec}$).
 5. Lower Tester expects an AUX_SCAN_RSP PDU from the IUT T_{IFS} after the AUX_SCAN_REQ.
 6. Repeat steps 4-5 100 times.
 7. The Upper Tester sends an HCI_LE_Set_Extended_Advertising_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
 8. If the IUT supports the LE Coded PHY, repeat steps 1-7, except in Step 1, the Primary_Advertising_PHY shall be set to 0x03 (LE Coded), and the Secondary_Advertising_PHY shall be set to 0x03 (LE Coded).
 9. If the IUT supports the LE 2M PHY, repeat steps 1-7, except in Step 1, the Primary_Advertising_PHY shall be set to 0x01 (LE 1M), and the Secondary_Advertising_PHY shall be set to 0x02 (LE 2M).
- Expected Outcome

Pass Verdict

The IUT responds to at least 95% of the AUX_SCAN_REQ packets sent by the Lower Tester in step 4.

The time between a PDU containing an AuxPtr field and the PDU to which it refers shall be greater than or equal to T_{MAFS} .

- Notes

Lower Tester is configured to use $1.5 \mu\text{sec}$ instead of $2 \mu\text{sec}$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.2 SLA

Tests that the IUT behaves according to timing constraints in the slave role. The slave timing requirements apply both in low power and active mode.

4.4.2.1 LL/TIM/SLA/BV-01-C [Adjusting Anchor Point]

- Test Purpose

Test that a slave IUT on accepting a parameter update from the master, adopts a new anchor point when starting to use the new parameters.

The Lower Tester acts in the master role in the connection, sending a connection parameter update packet to the IUT until it accepts it, then takes the new parameters into use varying the time to the earliest and latest possible. The Lower Tester observes the slave transmissions on the data channels used.

- Reference

[3] 5.1.1

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, 200 ms timeout, common channel map, any SCA value)



- Test Procedure

Execute the test procedure using a connection supervision timer parameter of 200 ms and zero slave latency. Update using the same connection parameters.

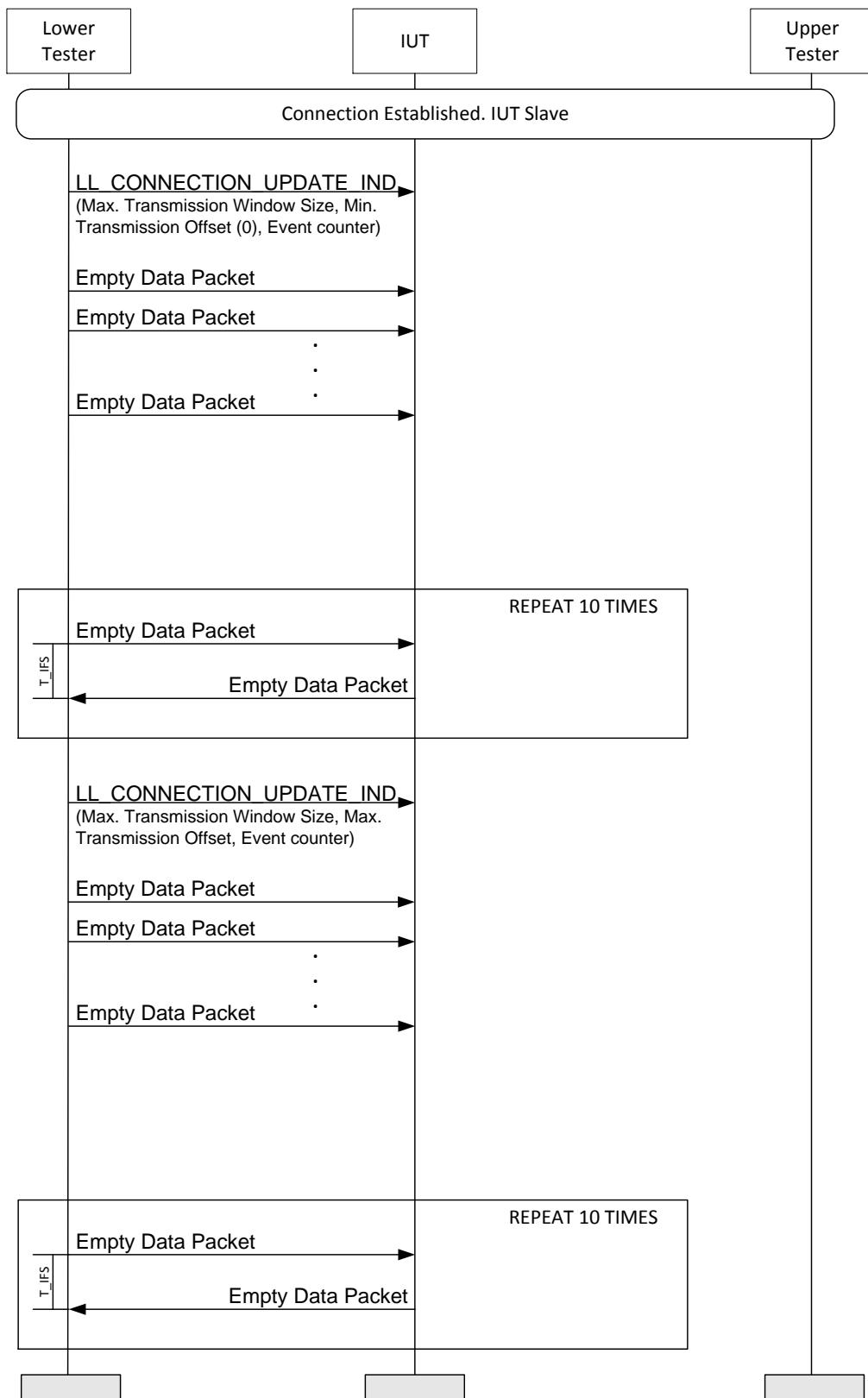


Figure 4.307: LL/TIM/SLA/BV-01-C [Adjusting Anchor Point]

1. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum Transmission Window Size and to the minimum Transmission Offset and expects a packet from the IUT acknowledging the connection update request.
 2. Lower Tester sends an empty DATA packet to the IUT and expects a response in all events until the event count matches the indicated time of connection update.
 3. Send an empty LL DATA packet starting the event at connection interval, using the common data channel selection parameters. Expect an empty LL DATA packet in response from the IUT, TIFS after the data packet sent on the same data channel. Repeat up to 15 times.
 4. Interleave with step 3: Expect no HCI_LE_Connection_Update_Complete event from the IUT.
 5. Lower Tester sends a CONNECTION_UPDATE_IND packet setting the connection parameters to the maximum Transmission Window Size and to the maximum Transmission Offset and expects a packet from the IUT acknowledging the connection update request.
 6. Repeat step 2.
 7. Repeat steps 3-4.
- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.
 - Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT adopting the new anchor point and maintaining the connection.

The IUT produces no HCI event reporting the connection update.
 - Notes

The calculations in the tests 'Earliest/Latest Transmission Start to Slave' can be used to define the connection parameters and test procedure details required. A difference in the parameter application is that the transmission window size is larger and the window offset is used.

4.4.2.2 LL/TIM/SLA/BV-02-C [Earliest Transmission Start to Slave]

- Test Purpose

Test that the slave IUT is able to establish and maintain a connection with a master that uses the earliest possible timing for the first transmission.

The Lower Tester acts first in the initiating state, then times the first transmission in the data channel at the earliest time possible.
- Reference

[3] 4.5.5
- Initial Condition

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

Execute the test using the common connection interval and data channel selection parameters, a connection supervision timer of 10 s and a zero slave latency in the connection request packet using



the minimum SCA for the most deviation by clock drift. Use a minimum transmission window size (1.25 ms) with 0 offset.

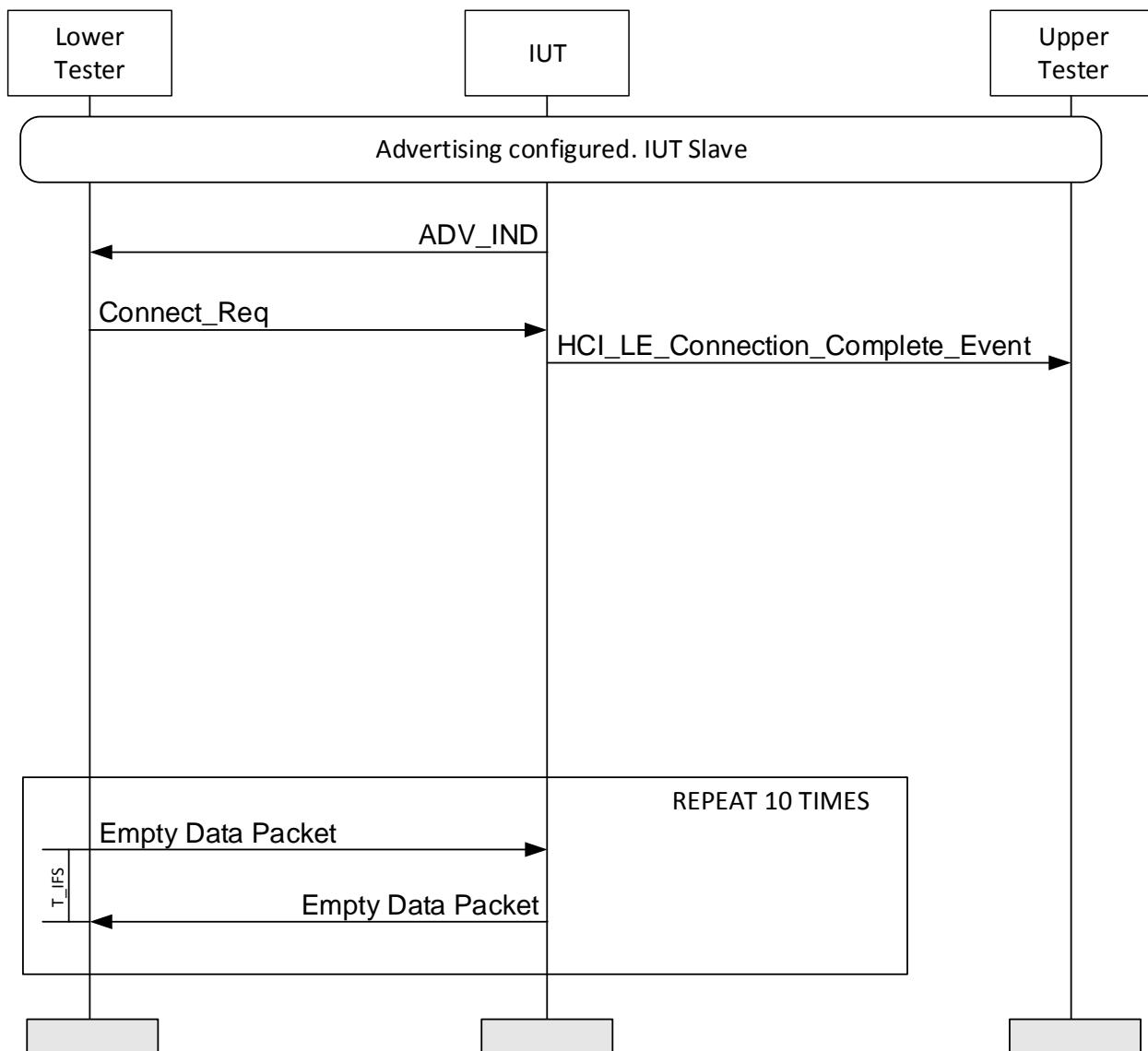


Figure 4.308: LL/TIM/SLA/BV-02-C [Earliest Transmission Start to Slave]

1. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel, respond with a CONNECT_IND packet T_IFS after the end of the advertising packet.
 2. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT with the parameters sent in step 1.
 3. Lower Tester sends an empty LL DATA packet starting the first event at connection interval transmitting at the leading edge of the window, using the common data channel selection parameters. Lower Tester expects an empty LL DATA packet in response from the IUT, T_IFS after the data packet sent on the same data channel. Repeat up to 10 times.
 4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 2).
- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.



- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT starting slave transmissions after the events skipped.

4.4.2.3 LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]

- Test Purpose

Test that the slave IUT is able to establish a connection with a master that uses the latest possible timing for the first transmission.

The Lower Tester acts first in the initiating state, then times the first transmission in the data channel at the latest time required.

- Reference

[3] 4.5.5

- Initial Condition

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

- Test Procedure

Execute the test, using the common connection interval and data channel selection parameters. Use a connection supervision timer of 690 ms, zero slave latency, the minimum SCA and a maximum transmission window and maximum transmission offset in the connection request packet.



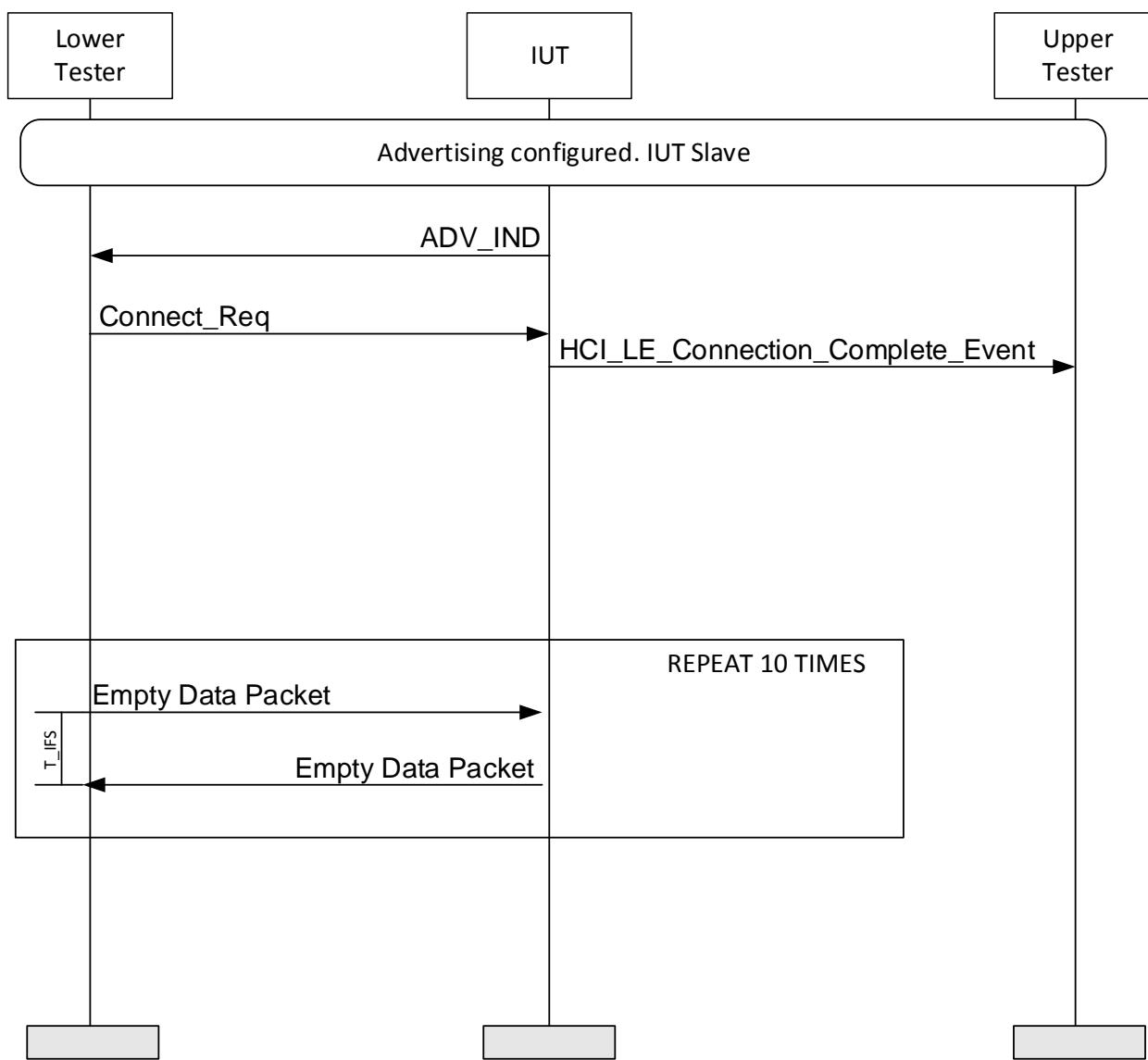


Figure 4.309: LL/TIM/SLA/BV-03-C [Latest Transmission Start to Slave]

1. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
 2. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT with the parameters sent in step 1
 3. Lower Tester sends an empty LL DATA packet starting the first event at connection interval transmitting at the trailing edge of the window, using the common data channel selection parameters.
 4. Lower Tester expects an empty LL DATA packet in response from the IUT, T_IFS after the data packet sent on the same data channel. Repeat up to 10 times.
 5. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 2).
- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.



- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT setting up the connection by starting slave transmissions after the skipped events.

4.4.2.4 LL/TIM/SLA/BV-04-C [Shortest Connection Interval]

- Test Purpose

Test that a slave IUT is able to maintain a connection with a master using the minimum timing between events (ConnInterval – 16μsec).

The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the minimum time required between events.

- Reference

[3] 4.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, -20 ppm SCA value)

- Test Procedure

Execute the test using the common data channel selection parameters, a nominal connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

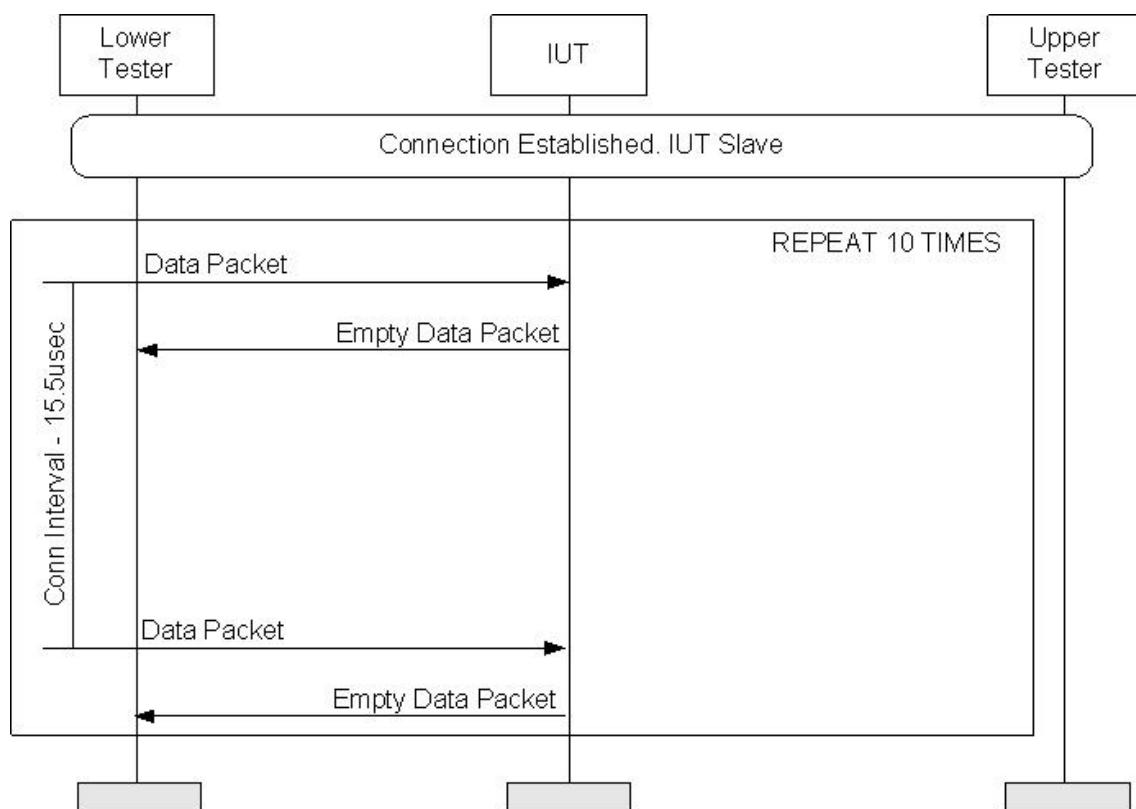


Figure 4.310: LL/TIM/SLA/BV-04-C [Earliest Transmission to Slave]



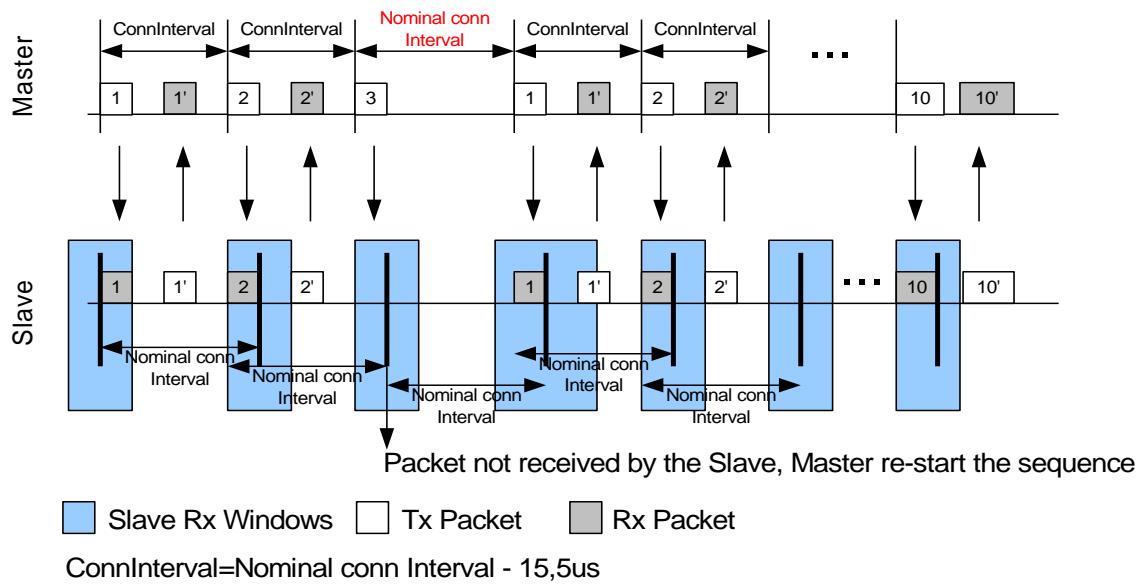


Figure 4.311: LL/TIM/SLA/BV-04 Timing sequence

1. Lower Tester sends DATA packets once a connection interval to the IUT is made using the data channel selection. Connection Interval will be set to nominal ConnInterval – 15.5 μ sec. Expect an empty DATA packet from the IUT T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Repeat up to 10 times. If the IUT does not respond to one Lower Tester packet, next packet sent by the Lower Tester will be at nominal ConnInterval, according to Figure 4.311.
 2. Repeat step 1 five times.
 3. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from preamble).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT maintaining the connection.

The IUT responds to all 10 packets sent by the Lower Tester in at least 3 of 5 repetitions.

- Notes

Lower Tester is configured to use 15.5 μ sec instead of 16 μ sec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

The parameters in this test are calculated for a BER of 0.1 % or better.

4.4.2.5 LL/TIM/SLA/BV-05-C [Longest Connection Interval]

- Test Purpose

Test that a slave IUT is able to maintain a connection with a master using the maximum timing between events (ConnInterval + 16 μ sec).

The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the maximum time required between events.

- Reference

[3] 4.2



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, 20 ppm SCA value).

- Test Procedure

Execute the test using the common data channel selection parameters, a nominal connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

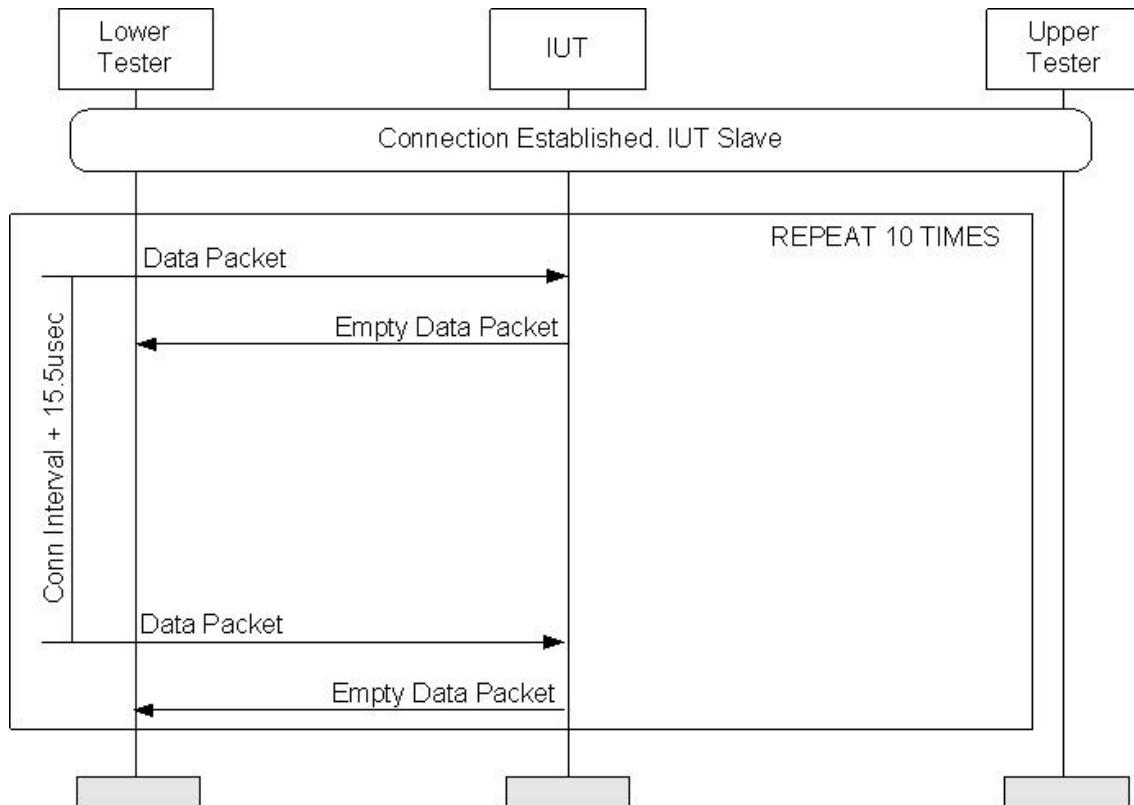


Figure 4.312: LL/TIM/SLA/BV-05-C [Latest Transmission to Slave]

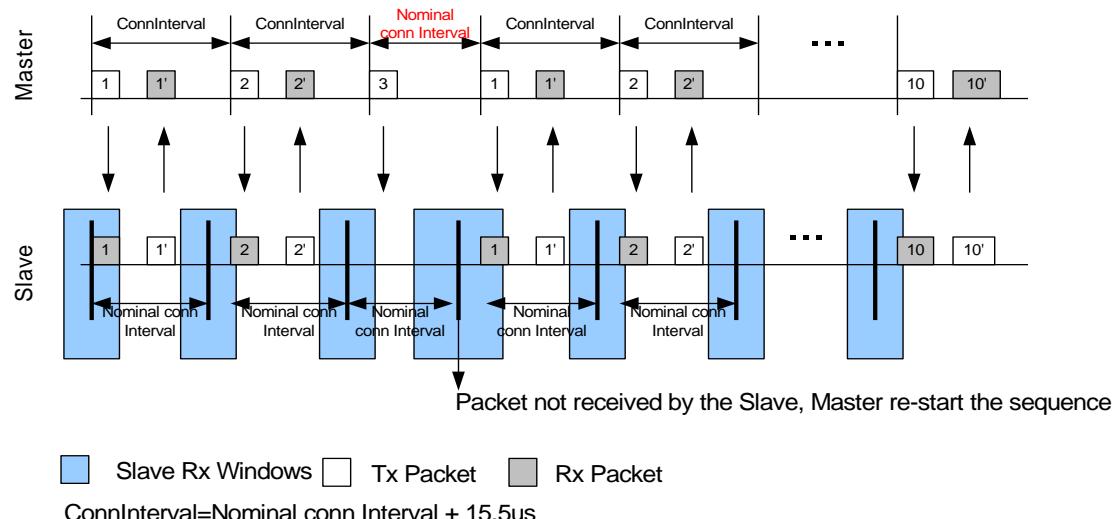


Figure 4.313: LL/TIM/SLA/BV-05-C Timing sequence



1. Lower Tester sends a DATA packet once a connection interval to the IUT, using the using the data channel selection parameters. Connection Interval will be set to nominal ConnInterval + 15.5 μ sec. Expect an empty DATA packet from the IUT T_IFS after each packet sent, with the SN matching the current NESN and the NESN matching the next SN. Repeat up to 10 times. If the IUT does not respond to one Lower Tester packet, next packet sent by the Lower Tester will be at nominal ConnInterval, according to [Figure 4.313](#).
 2. Repeat step 1 five times.
 3. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT maintaining the connection.

The IUT responds to all 10 packets sent by the Lower Tester in at least 3 of 5 repetitions.

- Notes

Lower Tester is configured to used 15.5 μ sec instead of 16 μ sec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

The parameters in this test are calculated for a BER of 0.1 % or better.

4.4.2.6 LL/TIM/SLA/BV-06-C [Earliest Transmissions to Slave]

- Test Purpose

Test that a slave IUT is able to maintain a connection with a master using the minimum timing between packets (T_IFS – 2 μ sec).

The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the minimum time required between packets.

- Reference

[3] 4.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure

Execute the test using the common data channel selection parameters, a connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.



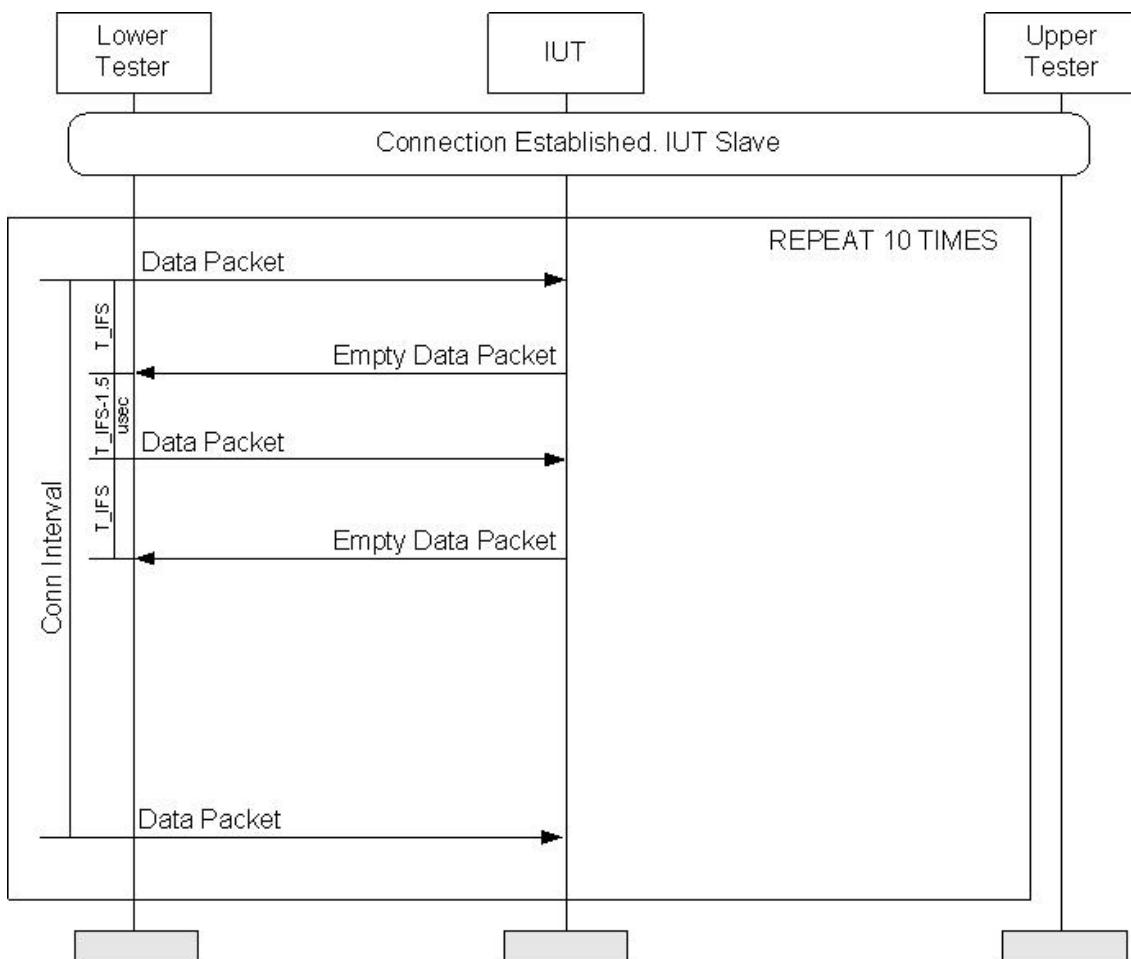


Figure 4.314: LL/TIM/SLA/BV-06-C [Earliest Transmissions to Slave]

1. Lower Tester sends a DATA packet to the IUT at the beginning of the Connection Event, setting MD bit to 1. Expect an empty DATA packet from the IUT TIFS after the packet sent.
 2. Lower Tester sends a second DATA packet in the same Connection Event, using the minimum time between packets ($T_{IFS} - 1.5 \mu\text{sec}$) and setting MD bit to 0. Expect an empty DATA packet from the IUT TIFS after the packet sent.
 3. Repeat up to 10 Connection Events.
 4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from preamble).
- Expected Outcome
- Pass Verdict
- The test procedure executes successfully, with the IUT maintaining the connection.
- The IUT responds at least to 95% of the packets sent by the Lower Tester in Step 2.
- Notes
- Lower Tester is configured to use $1.5\mu\text{sec}$ instead of $2\mu\text{sec}$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.2.7 LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]

- Test Purpose

Test that a slave IUT is able to maintain a connection with a master using the maximum timing between packets ($T_{IFS} + 2 \mu\text{sec}$).

The Lower Tester acts in the master role, maintaining the connection and sending data to the slave using the maximum time required between packets.

- Reference

[3] 4.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, selected connection interval, 0 common slave latency, selected timeout, common channel map, any SCA value)

- Test Procedure

Execute the test using the common data channel selection parameters, a connection interval of 10 ms and a connection supervision timer of 32 s in the connection request packet.

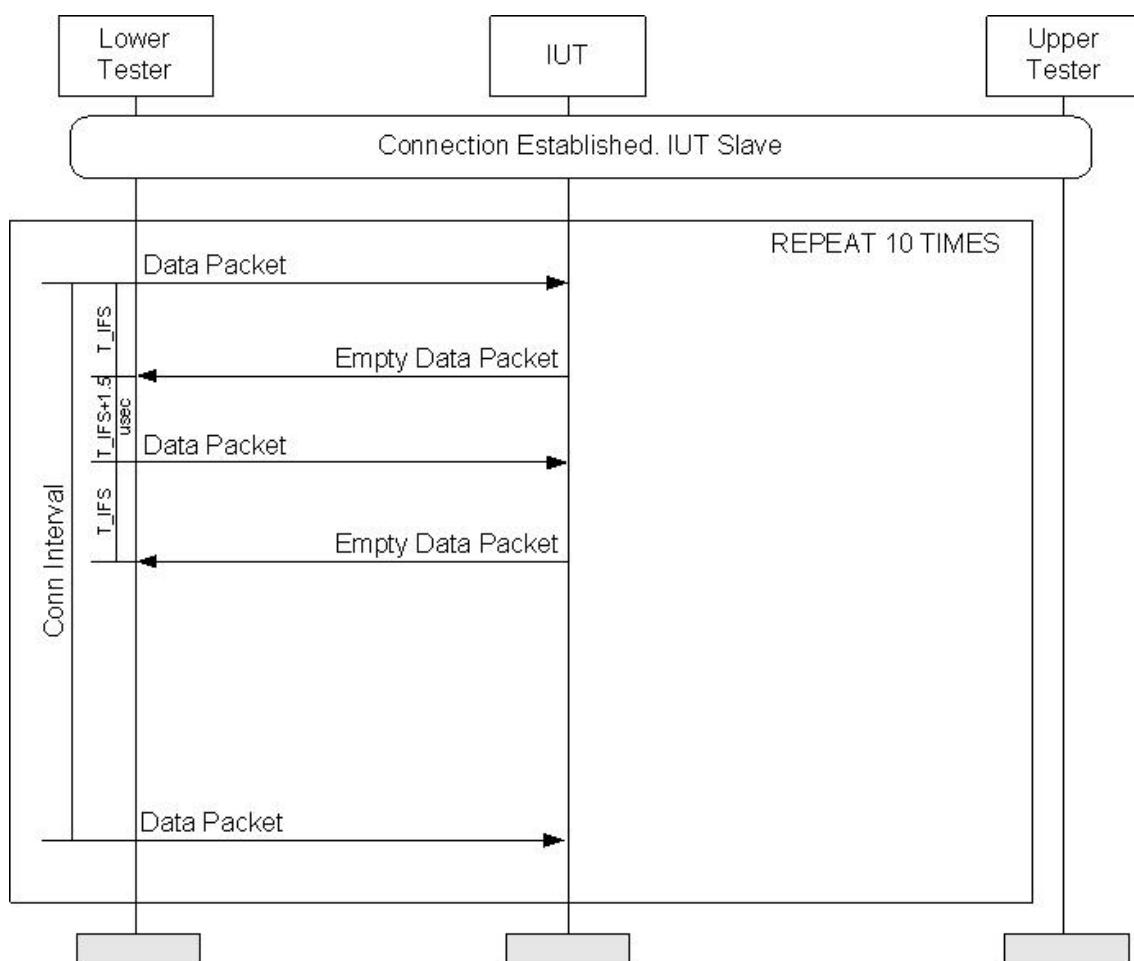


Figure 4.315: LL/TIM/SLA/BV-07-C [Latest Transmissions to Slave]



1. Lower Tester sends a DATA packet to the IUT at the beginning of the Connection Event, setting MD bit to 1. Expect an empty DATA packet from the IUT TIFS after the packet sent.
 2. Lower Tester sends a second DATA packet in the same Connection Event, using the maximum time between packets (TIFS + 1.5 μsec) and setting MD bit to 0. Expect an empty DATA packet from the IUT TIFS after the packet sent.
 3. Repeat up to 10 Connection Events.
 4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from preamble).
- Expected Outcome
- Pass Verdict
- The test procedure executes successfully, with the IUT maintaining the connection.
- The IUT responds at least to 95% of the packets sent by the Lower Tester in Step 2.
- Notes
- Lower Tester is configured to use 1.5μsec instead of 2μsec. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.2.8 LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]

- Test Purpose
- Test that the IUT initiates the Sleep Clock Accuracy Update procedure to change its Sleep Clock Accuracy. Verify that the connection remains active after the Sleep Clock Accuracy Update.
- Reference
- [13] 4.6.25, 5.1.14
- Initial Condition
- State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA and slave SCA values not the worst accuracy supported)



- Test Procedure

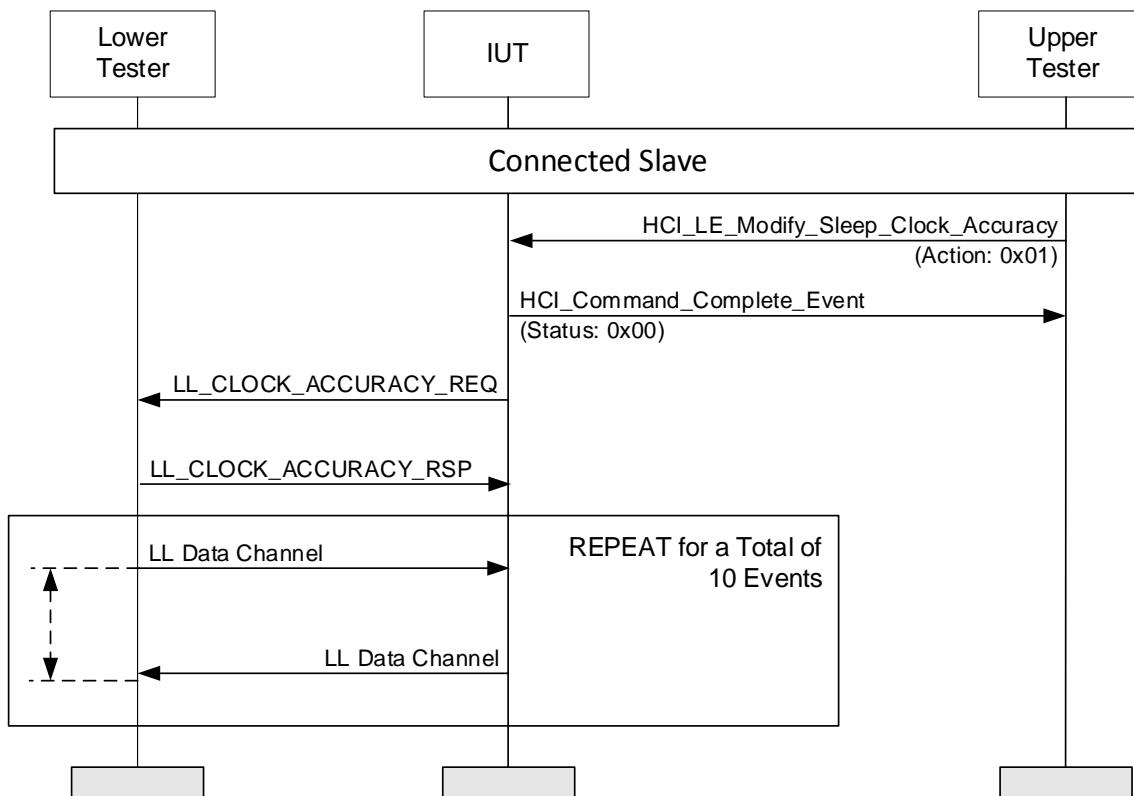


Figure 4.316: LL/TIM/SLA/BV-08-C [Initiate Sleep Clock Accuracy Update]

1. The Upper Tester sends an **HCI_LE_Modify_Sleep_Clock_Accuracy** command to the IUT with the Action parameter set to 0x01 and expects an **HCI_Command_Complete** event with zero status.
 2. The IUT sends an **LL_CLOCK_ACCURACY_REQ** PDU with the slaveSCA value that is worse than the initial slaveSCA from the connection process to the Lower Tester.
 3. The Lower Tester sends an **LL_CLOCK_ACCURACY_RSP** PDU to the IUT. The accuracy is worse than that currently in use.
 4. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
 5. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 6. Repeat for a number of events (10 events) to conclude that the connection is active.
- Expected Outcome

Pass Verdict

- The IUT receives 90% of the LL Data Channel PDUs from the Lower Tester in step 6 to conclude the connection is active.

4.4.2.9 LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update]

- Test Purpose

Test that the IUT responds to the Sleep Accuracy Update initiated by the Lower Tester. Verify that the connection remains active after the Sleep Clock Accuracy Update.



- Reference
[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, masterSCA value is 0 to 20ppm)

- Test Procedure

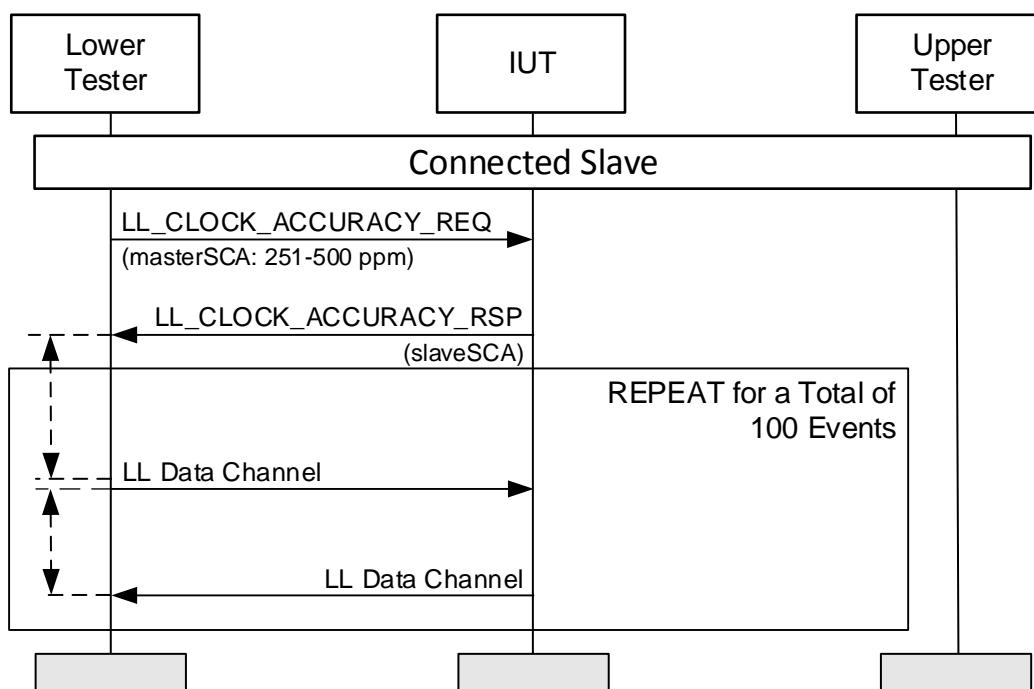


Figure 4.317: LL/TIM/SLA/BV-09-C [Response to Sleep Clock Accuracy Update]

1. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
2. The IUT receives the LL_CLOCK_ACCURACY_REQ PDU to update the Sleep Clock Accuracy. The IUT switches to the updated master Sleep Clock Accuracy value and sends the LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester.
3. After the Lower Tester receives the LL_CLOCK_ACCURACY_RSP PDU the Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel. The Lower Tester sends the packets within 2 µs of the edge of the timing window.
4. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
5. Repeat for a number of events (100 events) to conclude that the connection is active. For each event the Lower Tester randomly selects the leading edge or the trailing edge in step 3.

- Expected Outcome

Pass Verdict

- The IUT receives at least 95% of the LL Data Channel PDUs from the Lower Tester in step 5 to conclude the connection is active.



4.4.2.10 LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy]

- Test Purpose

Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester without reducing its Sleep Clock Accuracy.

- Reference

[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Slave (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, masterSCA value is 0 to 20ppm)

- Test Procedure

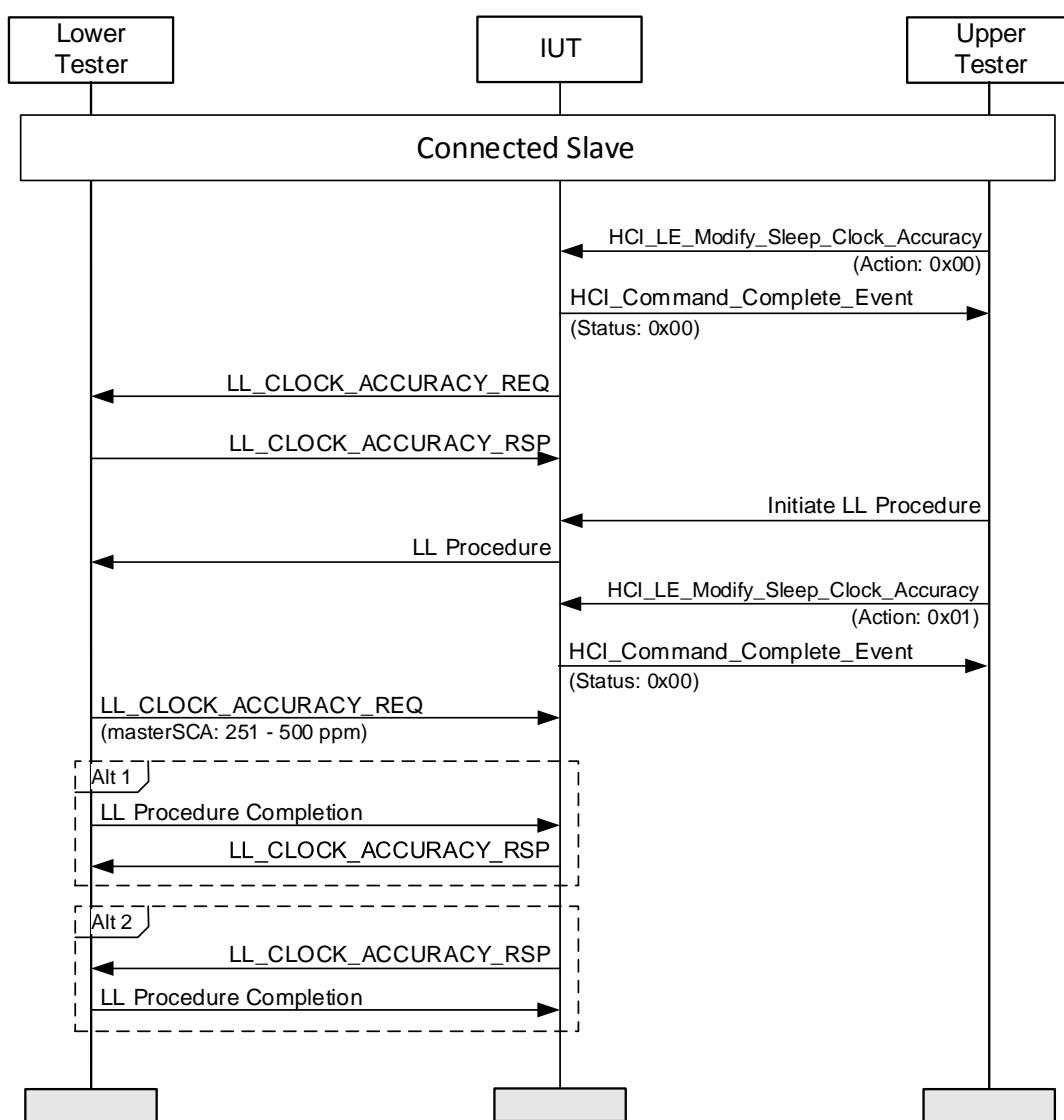


Figure 4.318: LL/TIM/SLA/BV-10-C [Response without Reducing the Sleep Clock Accuracy]



1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x00 and expects an HCI_Command_Complete event with status = 0x00 or 0x43.
 2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT.
 3. The Lower Tester responds with an LL_CLOCK_ACCURACY_RSP PDU.
 4. The Upper Tester sends a command that will cause the IUT to initiate an LL procedure with the Lower Tester. The LL procedure must not be the Sleep Clock Accuracy procedure and must be one that requires a response.
 5. The IUT sends the initial PDU of that procedure to the Lower Tester.
 6. The Upper Tester sends HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x01 and expects an HCI_Command_Complete event with zero status.
 7. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
 8. The IUT sends the LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester. If it has not done so within 20 connection intervals, the Lower Tester completes the procedure started in step 4 and then continues to wait for the LL_CLOCK_ACCURACY_RSP PDU.
- Expected Outcome
- Pass Verdict
- The LL_CLOCK_ACCURACY_RSP PDU sent by the IUT in step 8 has the same or better sleep clock accuracy than that in the LL_CLOCK_ACCURACY_REQ in step 2.

4.4.3 MAS

Tests that the IUT behaves according to timing constraints in the master role. The master timing requirements apply only in active mode.

4.4.3.1 LL/TIM/MAS/BV-01-C [Earliest Transmissions to Master]

- Test Purpose
Test that a master IUT is able to maintain a connection with a slave using the minimum inter-frame space in responses.
The Lower Tester acts in the slave role, maintaining the connection and responding to the master using the minimum time required to respond to packets from the master. The Lower Tester applies the active mode timing requirements by using the maximum negative jitter for the slave packet response transmissions.
- Reference
[\[3\] 4.2](#)
- Initial Condition
State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, common timeout, common channel map, not encrypted)
- Test Procedure
Execute the test procedure using the common connection interval, slave latency and connection supervision timer.



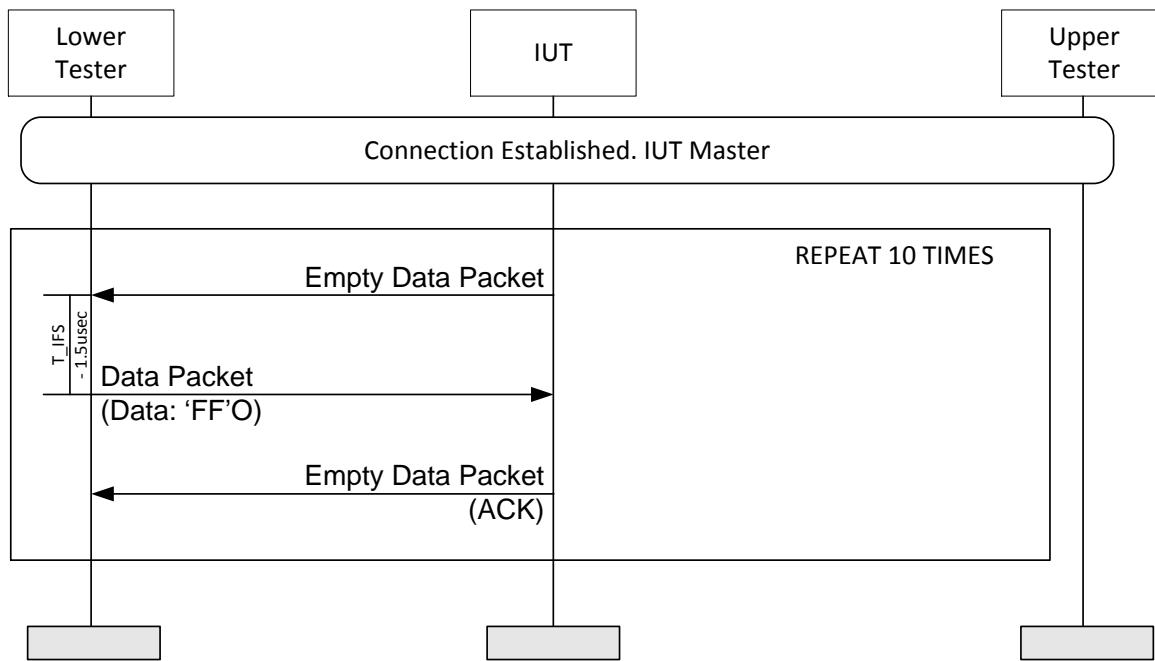


Figure 4.319: LL/TIM/MAS/BV-01-C [Earliest Transmission to Master]

1. Lower Tester expects an empty DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.
 2. In events where a response is required, Lower Tester sends a DATA packet in events required by the slave latency parameter only, using $T_{IFS}=150-1.5\mu\text{s}$ (maximum jitter value) and packet timing drift = -50 ppm (maximum positive clock drift) after the packet from the IUT.
 3. Repeat steps 1-2 15 times.
 4. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT maintaining the connection.

- Notes

Lower Tester is configured to use $1.5\mu\text{sec}$ instead of $2\mu\text{sec}$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.3.2 LL/TIM/MAS/BV-02-C [Latest Transmissions to Master]

- Test Purpose

Test that a master IUT is able to maintain a connection with a slave using the maximum inter-frame space in responses.

The Lower Tester acts in the slave role, maintaining the connection and responding to the master using the maximum time required to respond to packets from the master. The Lower Tester applies the active mode timing requirements by using the peak jitter for the slave packet response transmissions.

- Reference

[3] 4.2



- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, common connection interval, common slave latency, common timeout, common channel map, not encrypted)

- Test Procedure

Execute the test procedure using the common connection interval, slave latency and connection supervision timer.

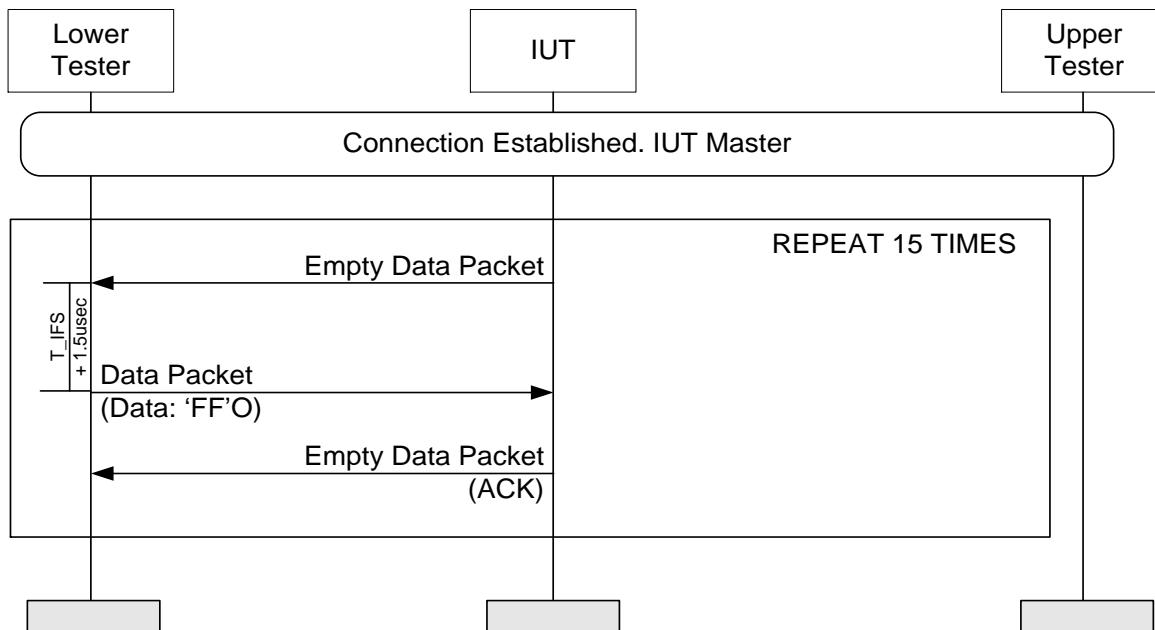


Figure 4.320: LL/TIM/MAS/BV-02-C [Latest Transmission to Master]

- Lower Tester expects an empty DATA packet once a connection interval from the IUT on the data channel derived from the selection parameters.
 - In events where a response is required, Lower Tester sends a DATA packet in events required by the slave latency parameter only, using $T_{IFS}=150+1.5\mu s$ (minimum jitter value) and packet timing drift = +50 ppm (maximum negative clock drift) after the packet from the IUT.
 - Repeat steps 1-2 15 times.
 - Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the initial state).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, IUT maintaining the connection.

- Notes

Lower Tester is configured to use $1.5\mu s$ instead of $2\mu s$. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.3.3 LL/TIM/MAS/BV-03-C [Initiate Sleep Clock Accuracy Update]

- Test Purpose

Test that the IUT initiates the Sleep Clock Accuracy Update procedure to change its sleep clock accuracy. Verify the accuracy of the timings of the connection event.



- Reference
[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value not the best accuracy supported)

- Test Procedure

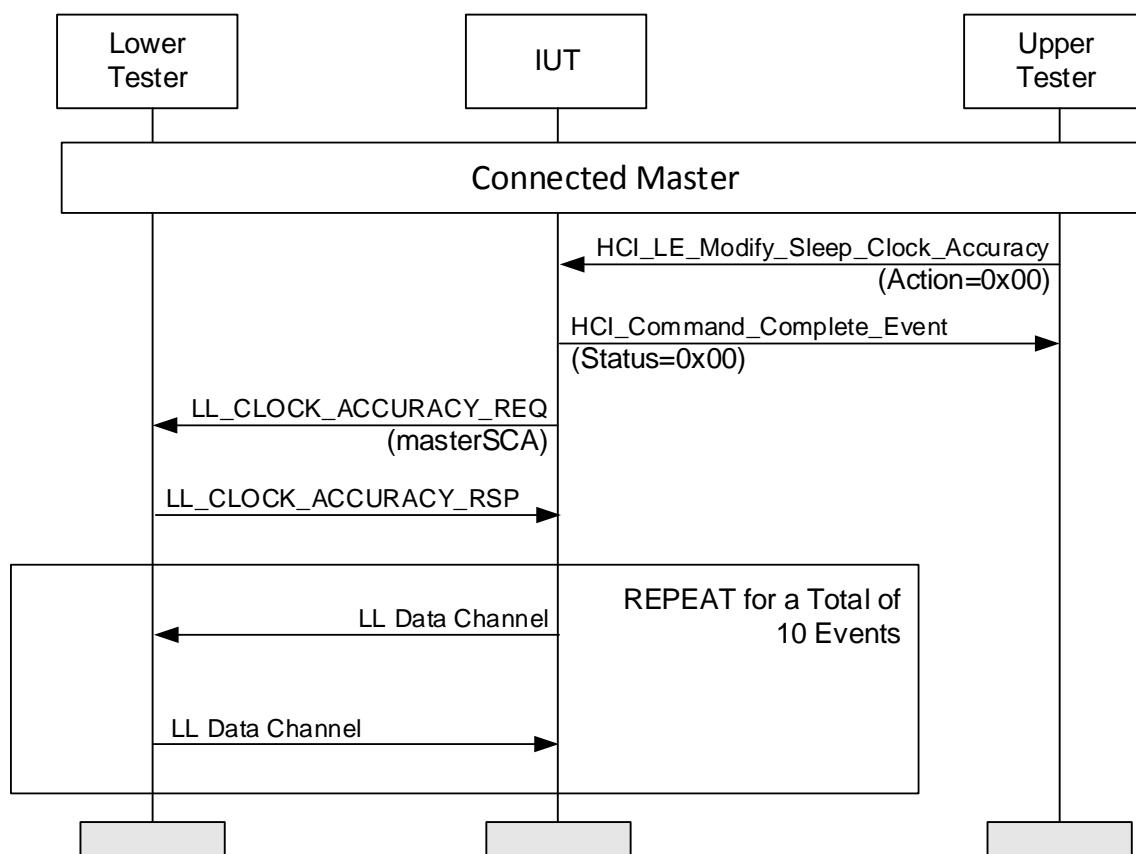


Figure 4.321: LL/TIM/MAS/BV-03-C [Initiate Sleep Clock Accuracy Update]

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action parameter set to 0x00 and expects an HCI_Command_Complete event with zero status.
2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the masterSCA value that is better than the initial slaveSCA from the connection process to the Lower Tester.
3. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT.
4. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
5. The Lower Tester sends a correctly formatted LL Data Channel PDU using the acknowledgement scheme, to the IUT on the same data channel.
6. Repeat for a number of events (10 events) to conclude the timing accuracy.



- Expected Outcome

Pass Verdict

- Verify that the connection event time intervals are within the timing range after the Sleep Clock Accuracy update.

4.4.3.4 LL/TIM/MAS/BV-04-C [Response to Sleep Clock Accuracy Update]

- Test Purpose

Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester. Verify the accuracy of the timings of the connection events.

- Reference

[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any master SCA value)

- Test Procedure

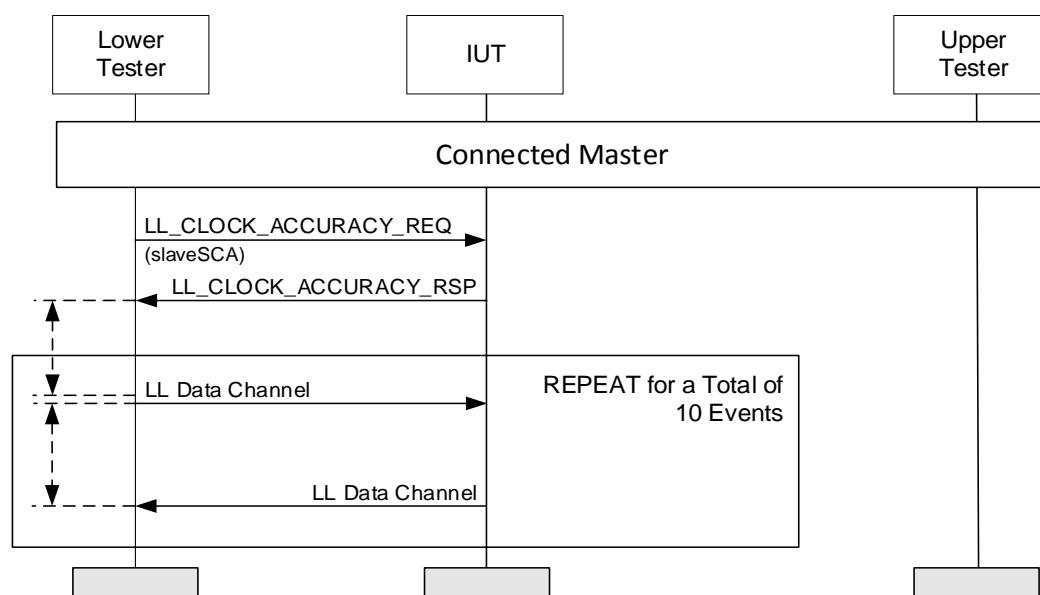


Figure 4.322: LL/TIM/MAS/BV-04-C [Response to Sleep Clock Accuracy Update]

1. The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT.
2. The IUT sends a LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester.
3. After the Lower Tester receives the LL_CLOCK_ACCURACY_RSP PDU the Lower Tester sends a correctly formatted LL Data Channel PDU, using the acknowledgement scheme, to the IUT on the same data channel.
4. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
5. Repeat for a number of events (10 events).



- Expected Outcome

Pass Verdict

- Verify that the connection event time intervals are within the timing range after the Sleep Clock Accuracy update.
- Verify that the Lower Tester receives the LL_CLOCK_ACCURACY_RSP with the master Sleep Clock Accuracy value.
- The LL_CLOCK_ACCURACY_RSP PDU sent by the IUT has the same or better sleep clock accuracy than the IUT was using at the start of the test.

4.4.3.5 LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy]

- Test Purpose

Test that the IUT responds to the Sleep Accuracy Update from the Lower Tester without reducing its Sleep Clock Accuracy.

- Reference

[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value is the best supported by the IUT)



- Test Procedure

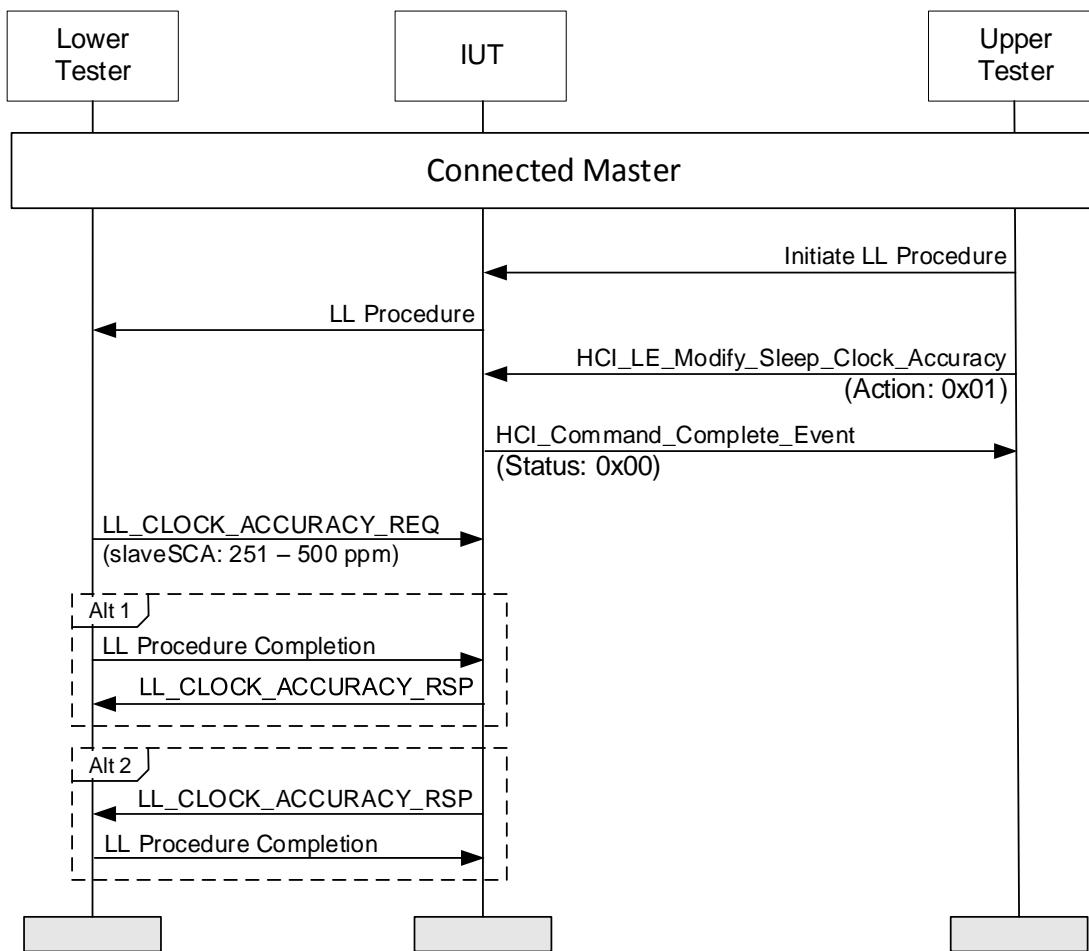


Figure 4.323: LL/TIM/MAS/BV-05-C [Response without Reducing the Sleep Clock Accuracy]

- The Upper Tester sends a command that will cause the IUT to initiate an LL procedure with the Lower Tester. The LL procedure must not be the Sleep Clock Accuracy procedure and must be one that requires a response.
- The IUT sends the initial PDU of that procedure to the Lower Tester.
- The Upper Tester sends HCI_Le_Modify_Sleep_Clock_Accuracy to the IUT with the Action parameter set to 0x01 and expects an HCI_Command_Complete event with zero status.
- The Lower Tester sends an LL_CLOCK_ACCURACY_REQ PDU to the IUT to update the sleep clock accuracy value to switch to a clock accuracy of 251 ppm to 500 ppm.
- The IUT sends the LL_CLOCK_ACCURACY_RSP PDU to the Lower Tester. If it has not done so within 20 connection intervals, the Lower Tester completes the procedure started in step 1 and then continues to wait for the LL_CLOCK_ACCURACY_RSP PDU.

- Expected Outcome

Pass Verdict

- The LL_CLOCK_ACCURACY_RSP sent by the IUT has the same or better sleep clock accuracy than the IUT was using at the start of the test.

4.4.3.6 LL/TIM/MAS/BV-06-C [Change Accuracy after Update]

- Test Purpose

Test that when the sleep clock accuracy gets worse the accuracy updates after the Lower Tester responds with the LL_CLOCK_ACCURACY_RSP PDU.

- Reference

[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value is the best supported by the IUT.)

- Test Procedure

Execute the test procedure using a connection interval of less than 500 ms.

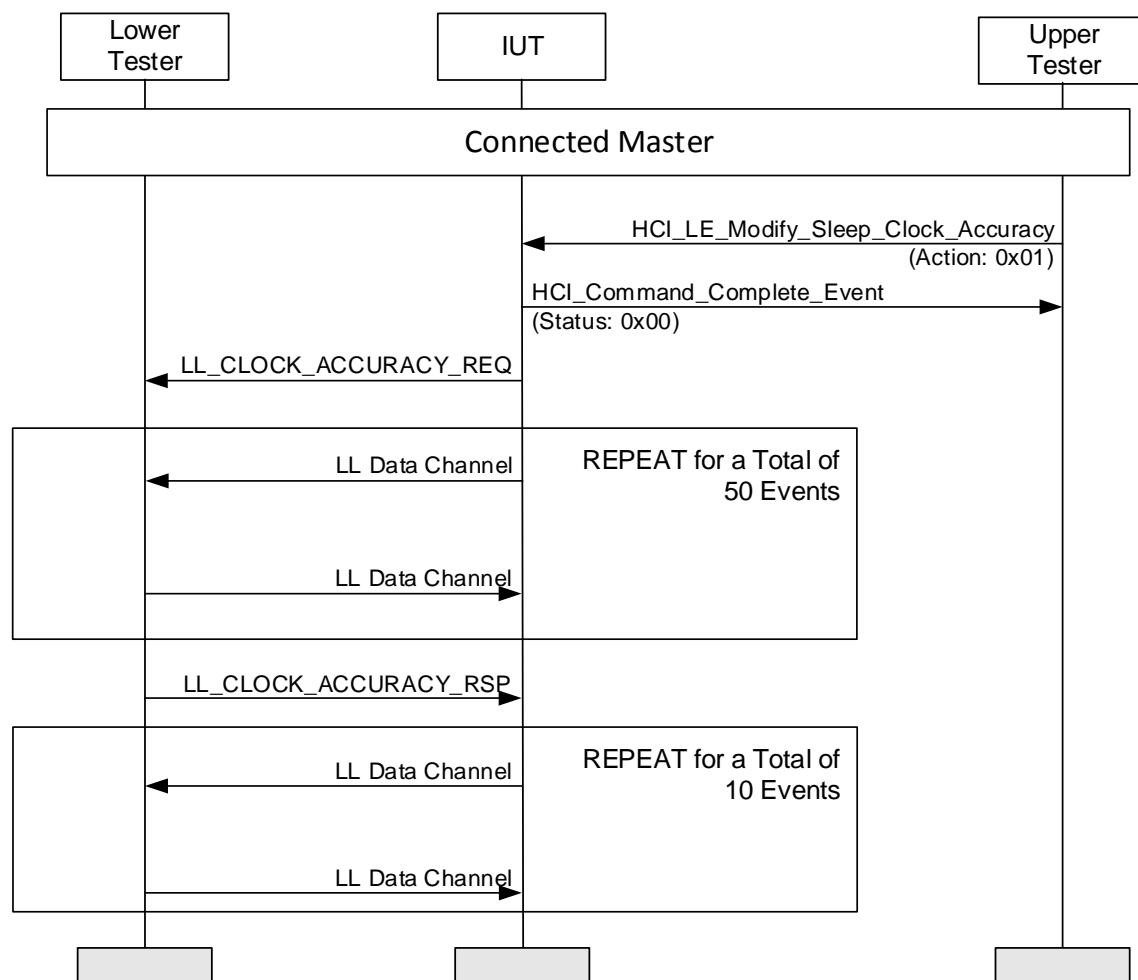


Figure 4.324: LL/TIM/MAS/BV-06-C [Change Accuracy after Update]

1. The Upper Tester sends an HCI_LE_Modify_Sleep_Clock_Accuracy command to the IUT with the Action Parameter set to 0x01 and expects an HCI_Command_Complete event with a zero status.

2. The IUT sends an LL_CLOCK_ACCURACY_REQ PDU with the masterSCA value set to a worse accuracy.
 3. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 4. Repeat for a number of events (50 events) to conclude that the timing accuracy has not changed.
 5. The Lower Tester sends an LL_CLOCK_ACCURACY_RSP PDU to the IUT.
 6. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 7. Repeat for a number of events (10 events).
- Expected Outcome
- Pass Verdict
- Verify that the timing of packets in steps 2 to 5 is no worse than the accuracy set in the initial condition.
 - Verify that the timing of packets in steps 6 and 7 are no worse than the accuracy specified in the LL_CLOCK_ACCURACY_REQ PDU.
 - The IUT receives at least 90% of the LL Data Channel PDUs from the Lower Tester in step 7 to conclude the connection is active.

4.4.3.7 LL/TIM/MAS/BV-07-C [Change Accuracy before Update]

- Test Purpose

Test that when the sleep clock accuracy gets better the sleep clock accuracy is updated before the Lower Tester sends the LL_CLOCK_ACCURACY_REQ PDU.

- Reference

[13] 4.6.25, 5.1.14

- Initial Condition

State: Connected Master (any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, master SCA value the worst supported by the IUT.)

- Test Procedure

Execute the test procedure using a connection interval of less than 500 ms.



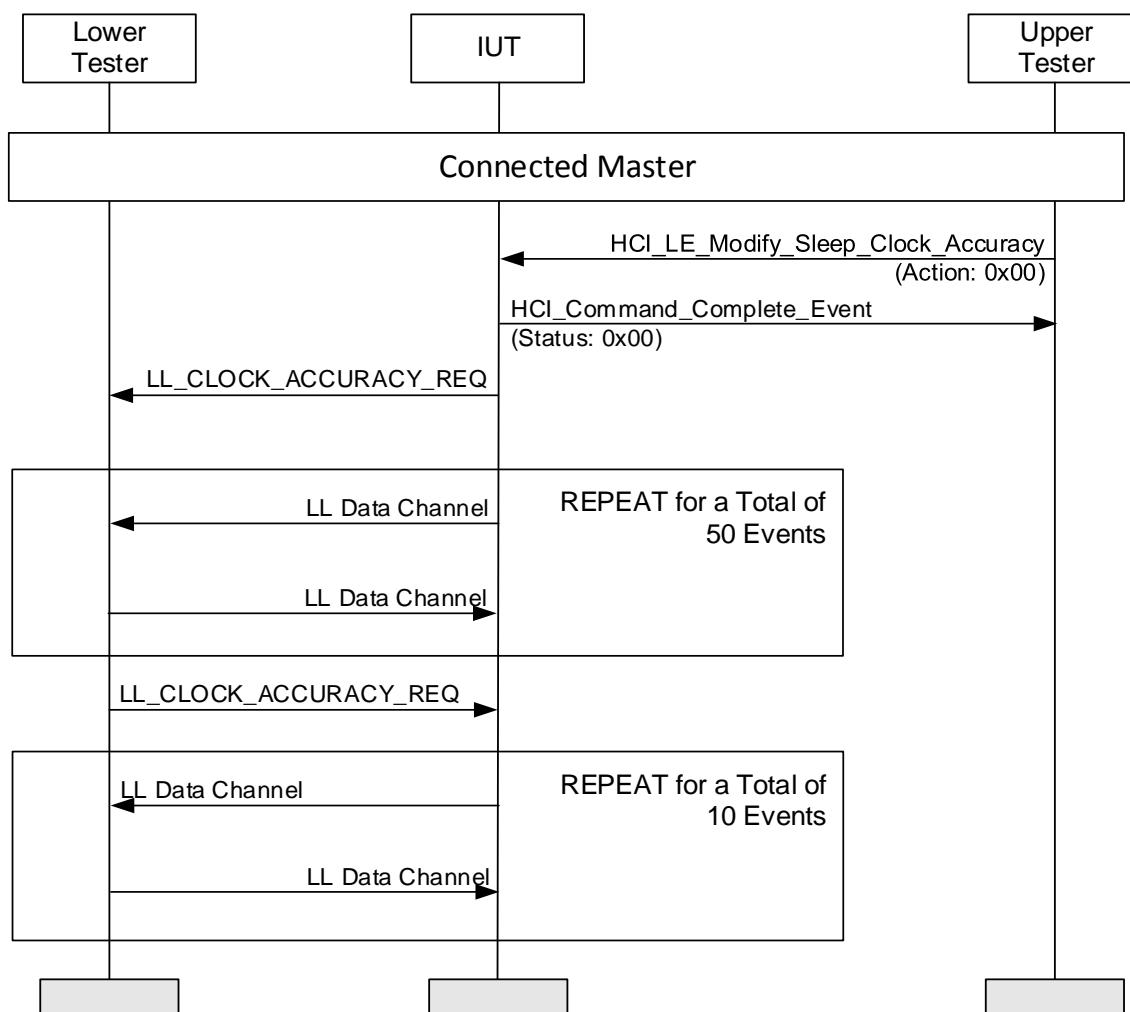


Figure 4.325: LL/TIM/MAS/BV-07-C [Change Accuracy before Update]

1. The Upper Tester sends an **HCI_LE_Modify_Sleep_Clock_Accuracy** command to the IUT with the Action Parameter set to 0x00 and expects an **HCI_Command_Complete** event with zero status.
 2. The IUT sends an **LL_CLOCK_ACCURACY_REQ** PDU with the masterSCA value set to a better accuracy.
 3. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 4. Repeat for a number of events (50 events) to conclude that the timing accuracy has changed.
 5. The Lower Tester sends an **LL_CLOCK_ACCURACY_RSP** PDU to the IUT.
 6. The Lower Tester expects correctly formatted LL Data Channel PDUs on subsequent data channels at connection intervals, calculated for the connection interval used.
 7. Repeat for a number of events (10 events).
- Expected Outcome
- Pass Verdict**
- Verify that the timing of packets in steps 3 to 7 is no worse than the accuracy specified in step 2.
 - The IUT receives at least 90% of the LL Data Channel PDUs from the Lower Tester in step 7 to conclude the connection is active.

4.4.4 SCN

Tests that the IUT behaves according to timing constraints as a scanner.

4.4.4.1 LL/TIM/SCN/BV-01-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner]

- Test Purpose

Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the minimum allowed timing (T_{MAFS}) on the secondary channel.

- Reference

[10] 4.4.2.5.4

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

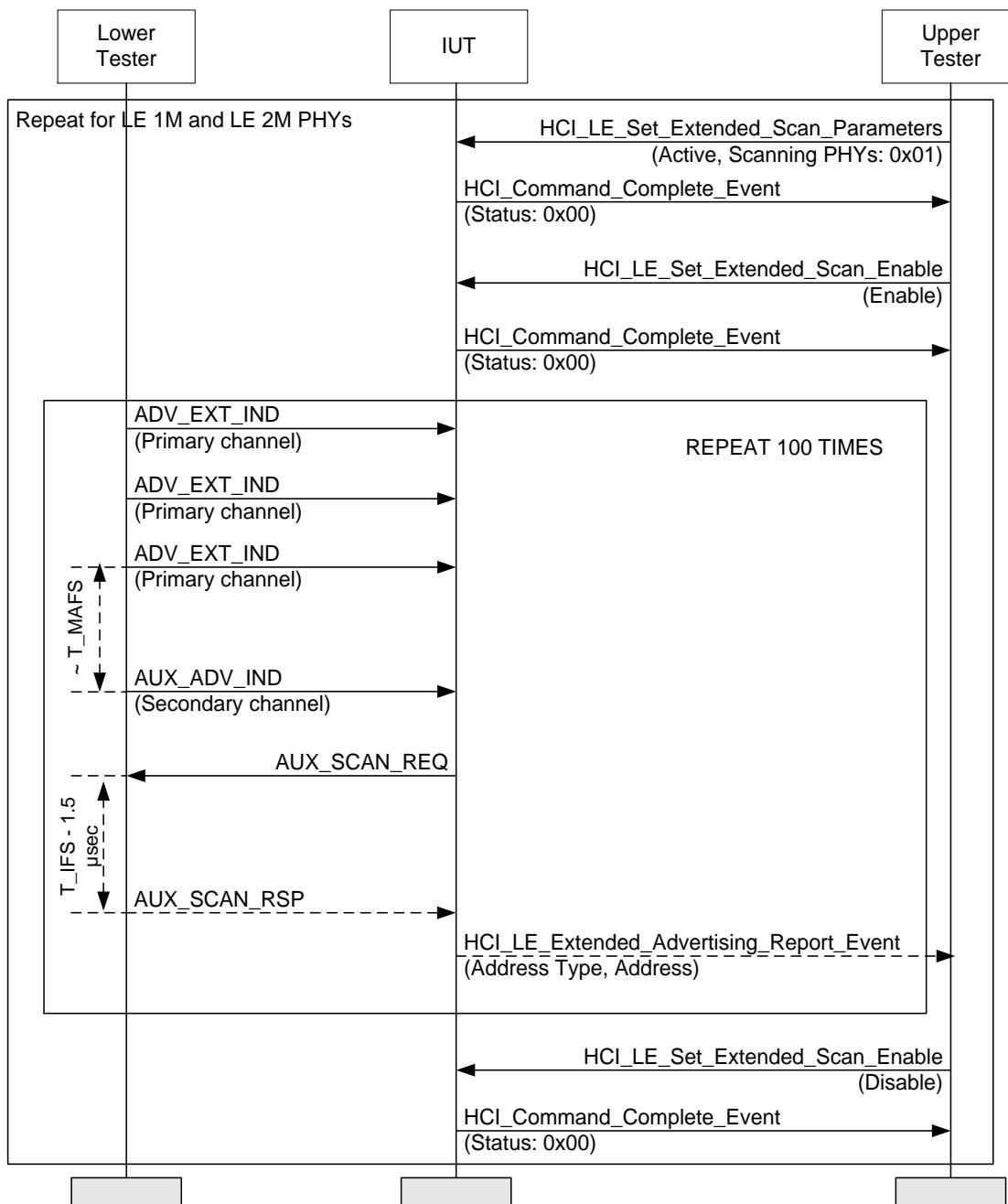


Figure 4.326: LL/TIM/SCN/BV-01-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner]

- Configure Lower Tester to start scannable advertising using the LE 1M PHY on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
- Upper Tester enables active scanning IUT with the Scanning PHYs field set to 0x01 (LE 1M PHY).
- Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the minimum time after the end of the ADV_EXT_IND PDU. The Aux Offset in the AuxPtr shall be the smallest representable value that is at least the length of the packet plus T_{MAFS} and the AUX_ADV_IND shall be transmitted at the represented time, which may be between 300 and 329 μs after the end of the packet.

4. The Lower Tester expects to receive an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the minimum time after the end of the AUX_SCAN_REQ PDU ($T_{IFS} - 1.5 \mu\text{sec}$).
 5. The Upper Tester expects one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
 6. Repeat steps 3-5 100 times.
 7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 8. If the IUT supports the LE 2M PHY, repeat steps 1-7, except in Step 1 the Lower Tester advertises using the LE 1M PHY for the primary advertising channel and the LE 2M PHY for the secondary advertising channel, and in Step 2 the Scanning_PHYs shall be set to 0x01 (LE 1M).
- Expected Outcome

Pass Verdict

The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.

4.4.4.2 LL/TIM/SCN/BV-02-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY]

- Test Purpose

Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the minimum allowed timing (T_{MAFS}) and AUX_SCAN_RSP PDUs when the PDU is sent at the minimum allowed timing ($T_{IFS} - 2 \mu\text{s}$) using the LE Coded PHY on the secondary channel.

- Reference

[10] 4.4.2.5.4

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

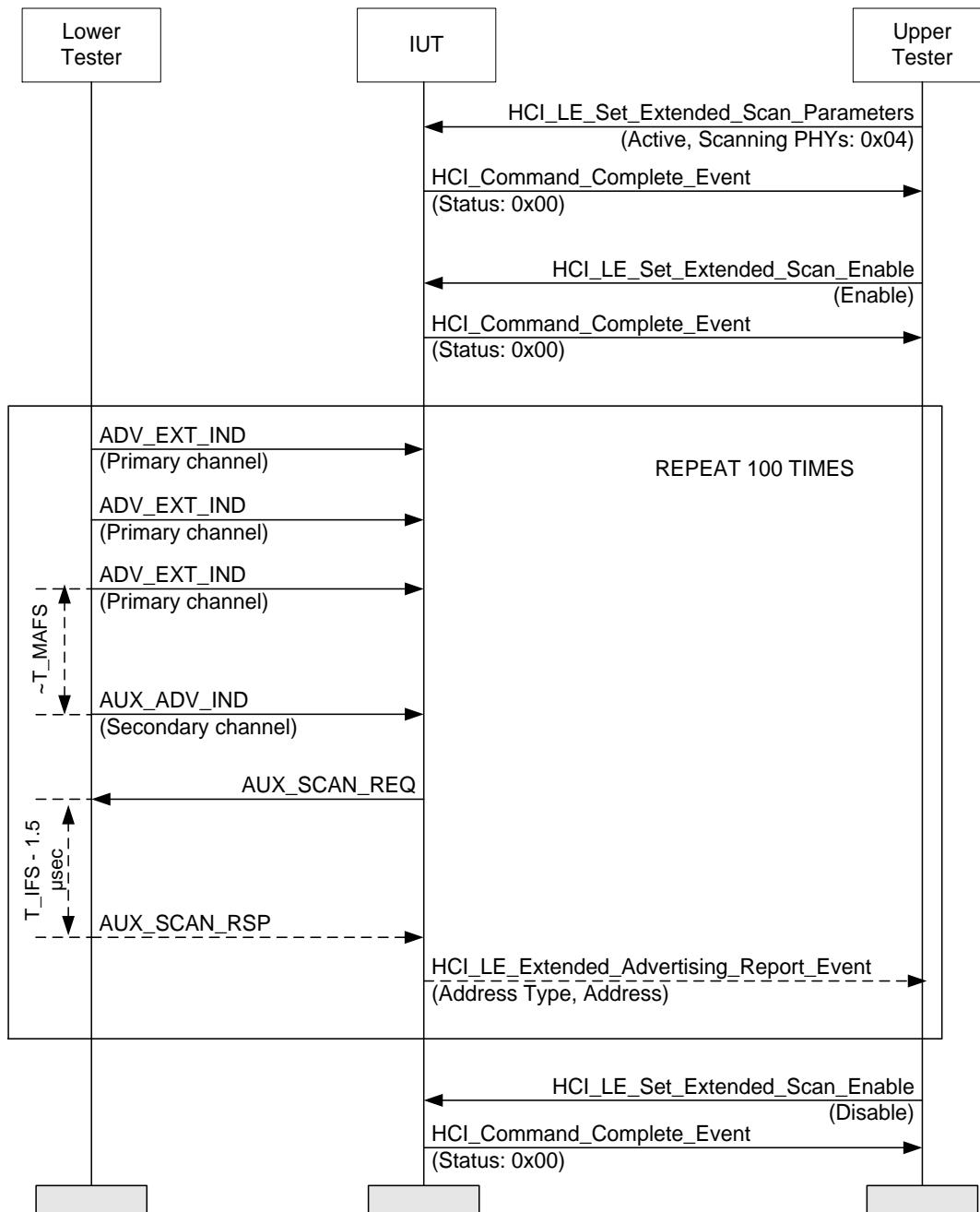


Figure 4.327: LL/TIM/SCN/BV-02-C [Extended Scanning, Secondary Channel, Earliest Transmission to Scanner, LE Coded PHY]

- Configure Lower Tester to start scannable advertising using the LE Coded PHY on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
- Upper Tester enables active scanning IUT with the Scanning PHYs field set to 0x04 (LE Coded PHY).
- Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the minimum time after the end of the ADV_EXT_IND PDU. The Aux Offset in the AuxPtr shall be the smallest representable value that

is at least the length of the packet plus T_MAFS and the AUX_ADV_IND shall be transmitted at the represented time, which may be between 300 and 329 μ s after the end of the packet.

4. The Lower Tester expects to receive an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the minimum time after the end of the AUX_SCAN_REQ PDU (T_IFS - 1.5 μ s).
5. The Upper Tester expects one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
6. Repeat steps 3–5 100 times.
7. Upper Tester sends an HCI_Command_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.

- Expected Outcome

Pass Verdict

The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.

- Notes

Lower Tester is configured to use 1.5 μ s instead of 2 μ s. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.4.4.3 LL/TIM/SCN/BV-03-C [Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY]

- Test Purpose

Tests that a scanner IUT can receive and process AUX_ADV_IND PDUs from an advertiser when the PDU is sent at the maximum allowed timing (Aux Offset + 1 Offset Unit) and AUX_SCAN_RSP PDUs when the PDU is sent at the maximum allowed timing (T_IFS + 2 μ s + simulated range delay) using the LE Coded PHY on the secondary channel.

- Reference

[10] 4.2.3, 4.4.2.5.4

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable Undirected Advertising (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

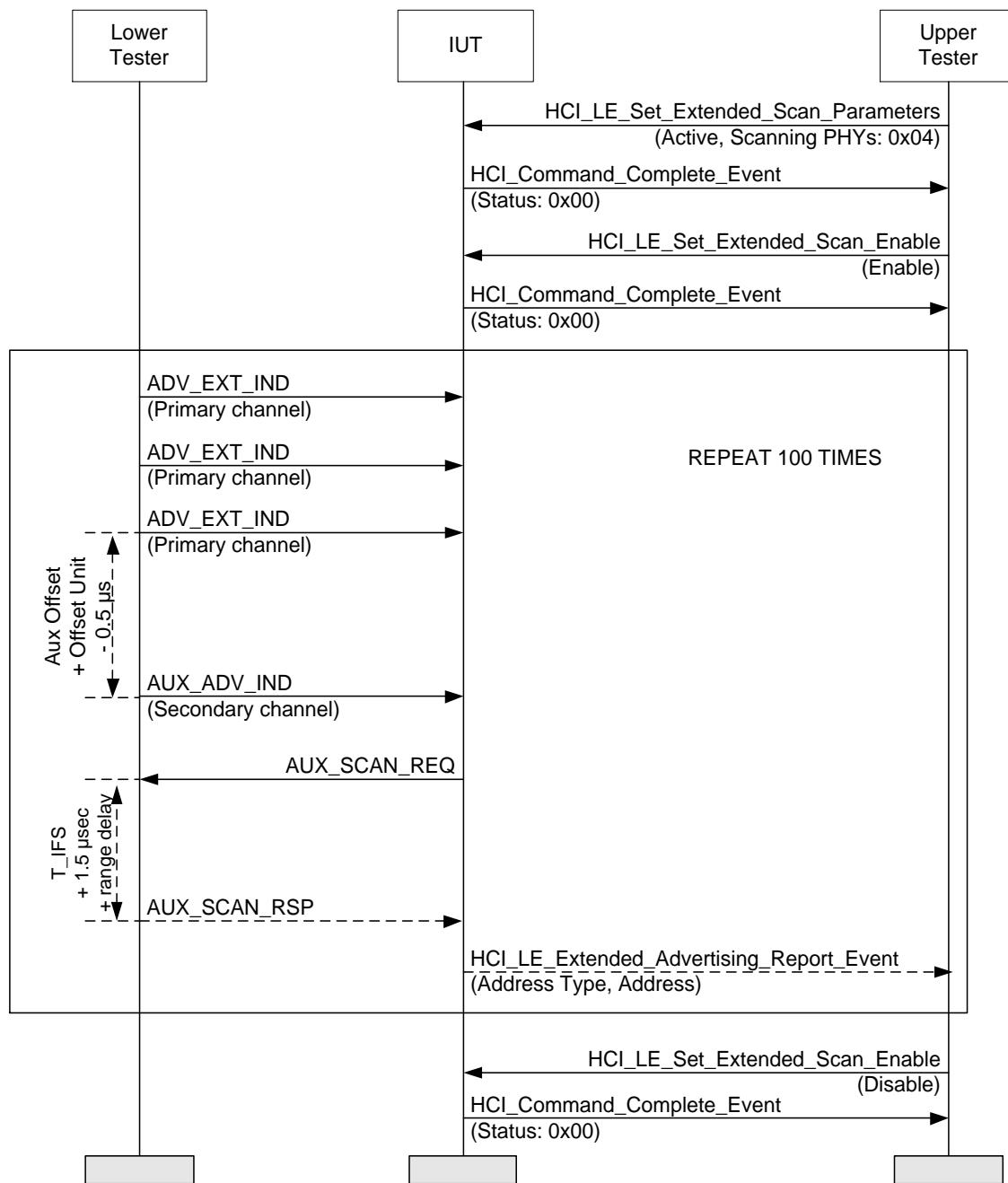


Figure 4.328: LL/TIM/SCN/BV-03-C [Extended Scanning, Secondary Channel, Latest Transmission to Scanner, LE Coded PHY]

- Configure Lower Tester to start scannable advertising using the LE Coded PHY on the primary and secondary advertising channels, with only channel 39 used for the primary advertising channel.
- Upper Tester enables active scanning IUT with the Scanning PHYs field set to 0x04 (LE Coded PHY).
- Lower Tester sends ADV_EXT_IND PDUs with an AuxPtr field referring to a valid AUX_ADV_IND PDU on the secondary advertising channel sent the maximum time after the end of the ADV_EXT_IND PDU (Aux Offset + Offset Unit - 0.5 μ s).

4. The Lower Tester expects to receive an AUX_SCAN_REQ PDU from the IUT and responds with an AUX_SCAN_RSP PDU the maximum time after the end of the AUX_SCAN_REQ PDU ($T_{IFS} + 1.5 \mu s + \text{simulated range delay}$), where simulated range delay is calculated as $2 * TSPX_{\text{scan_max_coded_range}} * 4 \text{ ns}$ with the result floored to the nearest multiple of $0.5 \mu s$ that is less than or equal to the result.
 5. The Upper Tester expects one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
 6. Repeat steps 3–5 100 times.
 7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
- Expected Outcome

Pass Verdict

The Upper Tester receives one or more HCI_LE_Extended_Advertising_Report_Events containing the Address_Type and Address of the Lower Tester with an Event_Type where bit 3 (Scan response) is set.
 - Notes

Lower Tester is configured to use $1.5 \mu s$ instead of $2 \mu s$ and Offset Unit - $0.5 \mu s$ instead of Offset Unit. This difference is needed to assure that there is no loss of synchronization due to Lower Tester timing resolution.

4.5 ENC

Tests that the IUT rejects packets with an invalid preamble or access address.

There is a possibility that the error injected in the access address is reverted on the radio path, for this the tests transmit the packet with the invalid access address more than once and typically reject behavior where the IUT responds over a particular count. Note that the pass criteria rely on a particular bit error rate for verdict assignment.

The access addresses used in the tests, while invalid data for the scenarios, are still formulated according to the access address formulation rules.

4.5.1 ADV

Tests that the IUT rejects packets with an invalid preamble or access address as an advertiser.

4.5.1.1 LL/ENC/ADV/BI-01-C [Scan Request Invalid Address]

- Test Purpose

Test that an advertiser IUT ignores a scan request with an invalid advertising channel synchronization word and continues advertising.

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

[3] 2.1.2, 4.4.2, 3.1



- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser)

- Test Procedure

Execute the test procedure with an advertising interval between the minimum and maximum advertising intervals.



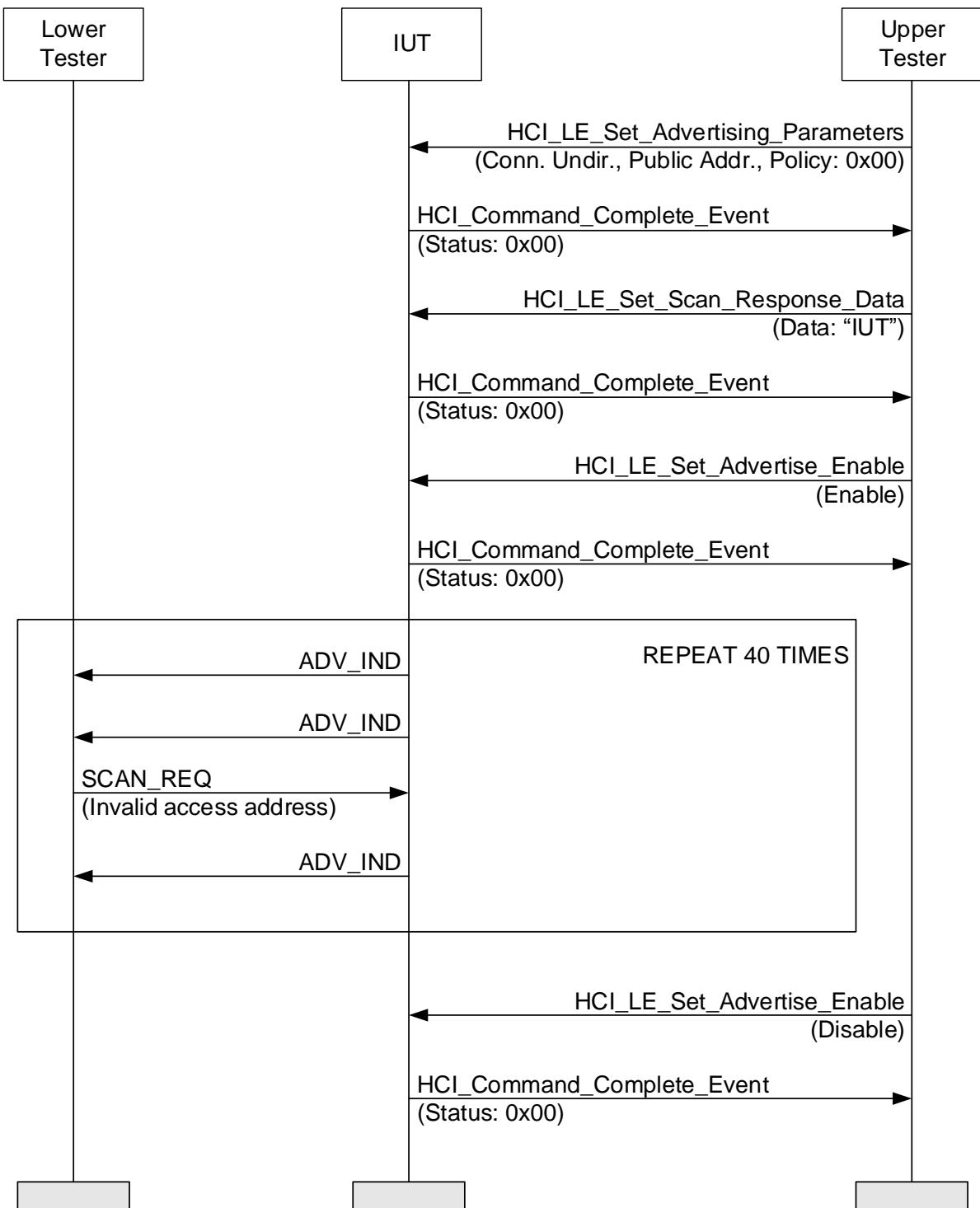


Figure 4.329: LL/ENC/ADV/BI-01-C [Scan Request Invalid Address]

1. Upper Tester enables undirected advertising in the IUT using all supported advertising channels, minimum advertising intervals and filtering policy set to 'Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)' if filtering policy is supported.
2. Configure Lower Tester to monitor the advertising and scan response procedures of the IUT.
3. Lower Tester sends a **SCAN_REQ** packet on a selected supported advertising channel (defined as an IXIT) and using a White Listed device address as parameter with an invalid access address from corrupting a single bit in the access address after receiving an **ADV_IND** packet from IUT on

the advertising channel configured in step 2. The SCAN_REQ packet is sent T_IFS after the end of an ADV_IND packet.

4. Lower Tester expects the IUT to continue advertising with no response to the SCAN_REQ packet.
5. Repeat steps 3-4 40 times, each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 SCAN_REQ packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 in that order; for the other 20 SCAN_REQ packets, it shall corrupt bits 31 to 12 in that order.
6. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to the IUT to disable advertising and expects an HCI_Command_Complete event in response.

- **Test Condition**

The parameters in this test are calculated for a BER of 0.1 % or better.

- **Expected Outcome**

Pass Verdict

The test procedure completes with the IUT responding to not more than 8 of the 40 SCAN_REQ packets sent by the Lower Tester in step 3.

- **Notes**

This test verifies the reverse of a positive requirement.

The count of responses (more than 8 out of 40) representing the single bit error reverting and the rest of the packet intact is calculated not to occur in practice using a binomial distribution.

Bit 0 is the first bit transmitted over the air.

4.5.1.2 LL/ENC/ADV/BI-02-C [Master Packets Invalid Address]

- **Test Purpose**

Test that an advertiser IUT receives a connection request, stops advertising after reception and after transmissions with an access address different from the connection request from the master until the connection supervision timer expires, considers the connection setup failed.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role but with packets with a mismatching access address.

- **Reference**

[3] 2.1.2, 4.4.4, 3.1

- **Initial Condition**

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map.

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser)

- **Test Procedure**

Execute the test procedure using the common data channel selection parameters, zero slave latency and a connection interval one tenth of the connection supervision timeout value (10 ms and 100 ms) in the connection request.



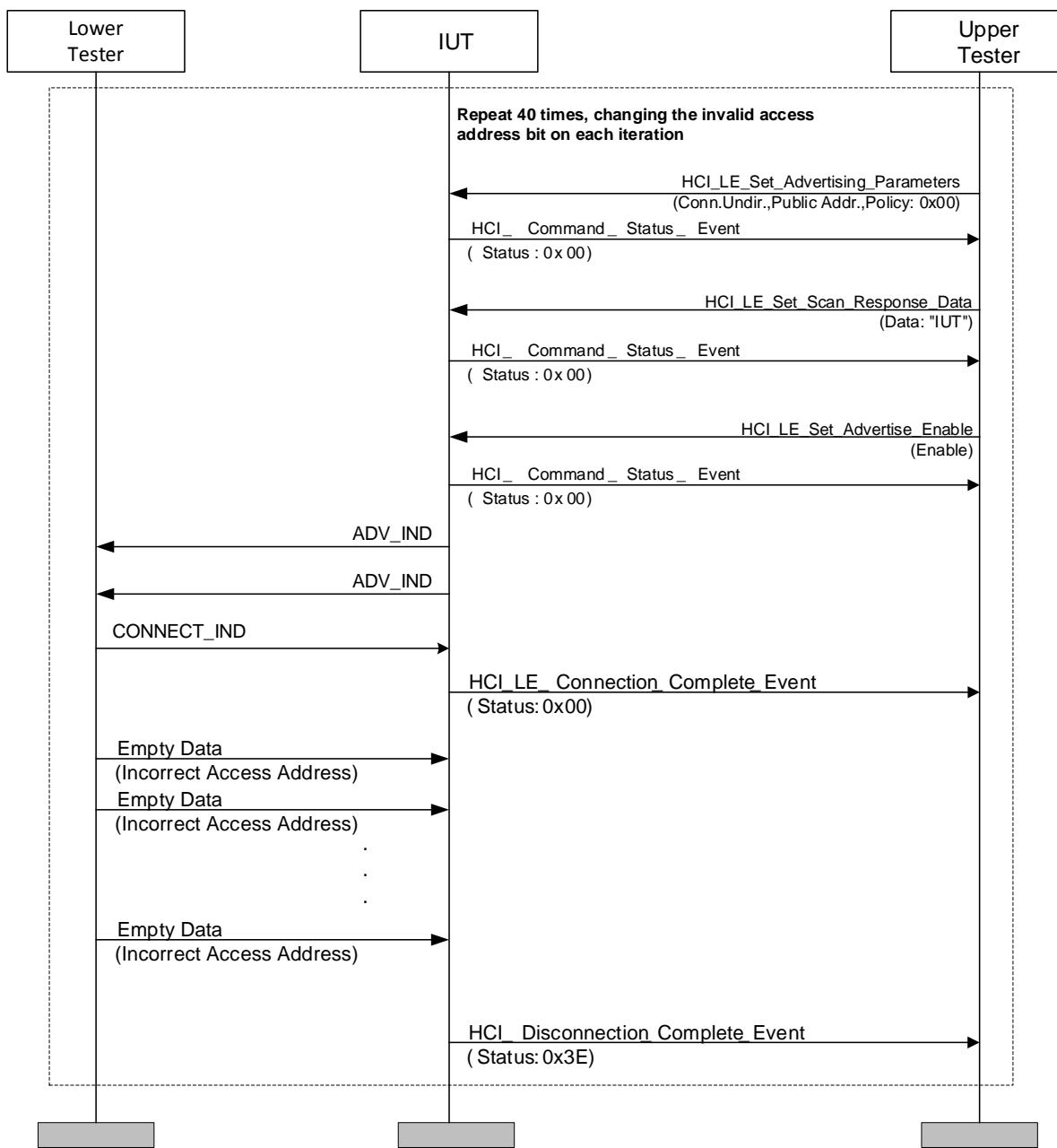


Figure 4.330: LL/ENC/ADV/BI-02-C [Master Packets Invalid Address]

1. Upper Tester enables undirected advertising in the IUT using one selected advertising channel (defined as an IXIT) and filtering policy set to 'Allow Scan Request from Any, Allow Connect Request from Any (Default) (0x00)' if filtering policy is supported.
2. Configure Lower Tester to initiate a connection.
3. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet T_IFS after the end of the advertising packet.
4. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.
5. Lower Tester sends an empty DATA packet with an access address different from the connection request (corrupting a single bit), starting the first event using the common connection interval timing after the connection request.
6. Lower Tester expects no response from the IUT.

7. Repeat steps 5-6 until a time period equal to twice the connection supervision timeout value or until step 8 executes:
 8. Upper Tester expects an HCI_Disconnection_Complete event indicating that creation of the connection failed (connection has failed to be established), with the connection handle parameter matching to step 4.
 9. Repeat steps 1-8 40 times, each time changing the position of the bit which is corrupted in the access address as follows: In the first 20 iterations, the Lower Tester shall corrupt bits 1 to 20 in that order; for the other 20 iterations, it shall corrupt bits 31 to 12 in that order.
- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.
 - Expected Outcome

Pass Verdict

For each of the 40 connection attempts, the test procedure completes with the IUT stopping advertising, moving to the data channel and reporting the failure to maintain the connection. The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete_Event with reason code 0x3E is received in at least 32 of 40 connection attempts.
 - Notes

The advertising channel used for the connection setup is assumed not to influence the access address recognition in the data channels.

The probability calculations describing the range of responses to expect are described in the test [LL/ENC/SLA/BI-01-C \[Packets to another Slave\]](#).

4.5.2 SCN

Tests that the IUT rejects packets with an invalid preamble or Access Address as a scanner.

4.5.2.1 LL/ENC/SCN/BI-01-C [Passive Scanning Invalid Address]

- Test Purpose

Test that a passive scanner IUT ignores advertising indication packets with invalid channel addresses.

The Lower Tester sends advertising packets with an invalid advertising channel synchronization word and checks that the IUT does not react to them.
- Reference

[3] 2.1.2, 4.4.3.1, 3.1
- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Passive Scanning (selected scan interval, selected scan window AND (All White Listed (policy for scanner)
- Test Procedure

Execute the test procedure with a timing combination using the minimum scan interval and maximum scan window supported, such that the scan interval is 3 times the length of the average advertising



interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).

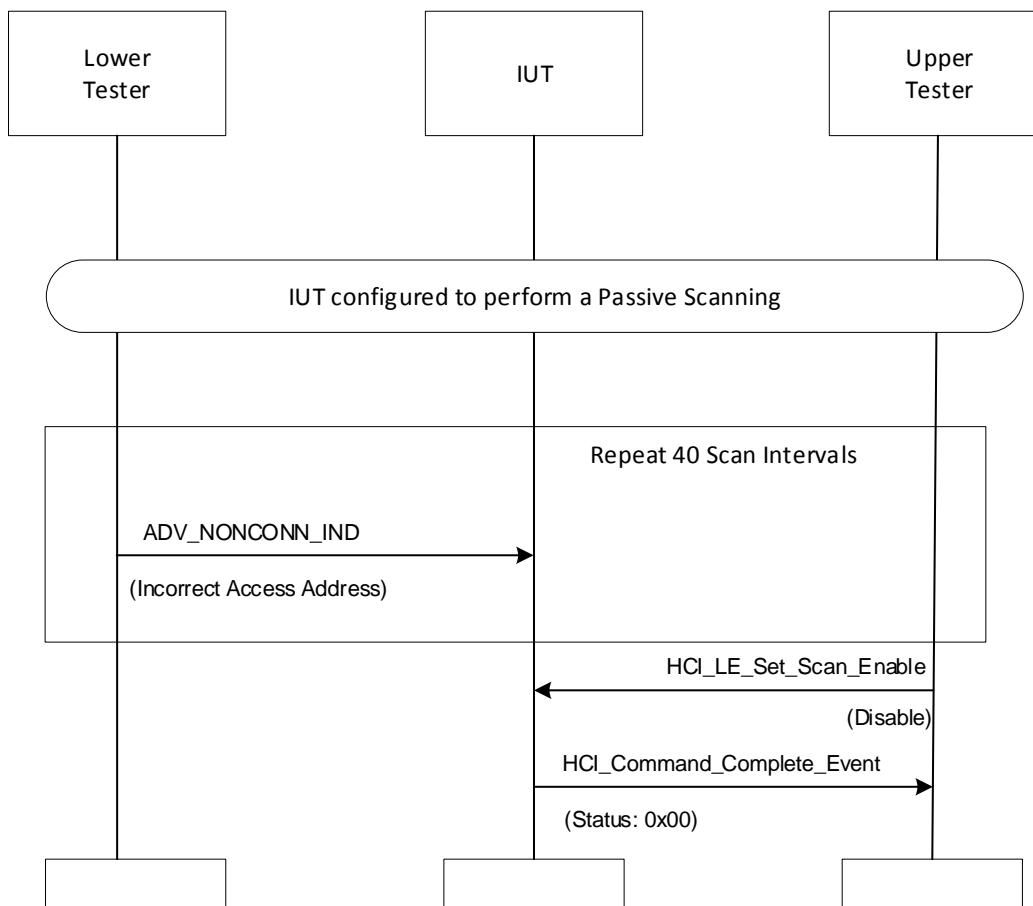


Figure 4.331: LL/ENC/SCN/BI-01-C [Passive Scanning Invalid Address]

1. Upper Tester enables passive scanning in the IUT.
 2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), and a white listed device address.
 3. Lower Tester sends an `ADV_NONCONN_IND` packet with an invalid access address, each advertising event using the selected advertising channel only, timing the events to match the scan interval.
 4. Upper Tester expects no `HCI_LE_Advertising_Report` reporting the advertising packets sent by the Lower Tester.
 5. Repeat steps 3-4 for a time that exceeds a number of scan intervals (40), each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 `ADV_NONCONN_IND` packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 of the access address in that order. For the other 20 `ADV_NONCONN_IND` packets, it shall corrupt bits 31 to 12 of the access address in that order.
 6. Upper Tester sends an `HCI_LE_Set_Scan_Enable` to the IUT to stop the scanning function and expect an `HCI_Command_Complete` event in response.
- Test Condition

The parameters in this test are calculated for a BER of 0.1 % or better.

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT reporting not more than 8 of the 40 advertising packets sent in step 3.

- Notes

Bit 0 is the first bit transmitted over the air.

The possibility for a scanner device to respond to an invalid radio frame (probability calculations described in the test [LL/ENC/SLA/BI-01-C \[Packets to another Slave\]](#)) provide for the case that an error injected into the access address is reverted. While not likely, such a bit error may occur in test execution, which would make a test not accepting any scan reports unreliable. Because the probability for the scanner to be listening the frequency of the advertising packet is unknown, this is ignored in the calculations. This test fails implementations not checking all the bits in the access address.

4.5.2.2 [LL/ENC/SCN/BI-02-C \[Active Scanning Invalid Address\]](#)

- Test Purpose

Test that an active scanner IUT ignores advertising indication packets with invalid channel addresses.

The Lower Tester sends advertising packets with an invalid access address and checks that the IUT does not react to them.

- Reference

[\[3\]](#) 2.1.2, 4.4.3.2, 3.1

- Initial Condition

Parameters: LL_scanner_scanInterval_MIN, LL_scanner_scanInterval_MAX, LL_scanner_scanWindow_MIN, LL_scanner_scanWindow_MAX, LL_scanner_Adv_Channel_Map.

State: Active Scanning (public address, selected scan interval, selected scan window) AND (All White Listed (policy for scanner))

- Test Procedure

Execute the test procedure using a scan interval 3 times the length of the average advertising interval. If device filtering is supported, use the filtering policy to white list all unknown devices (accept all advertising packets (0x00)).



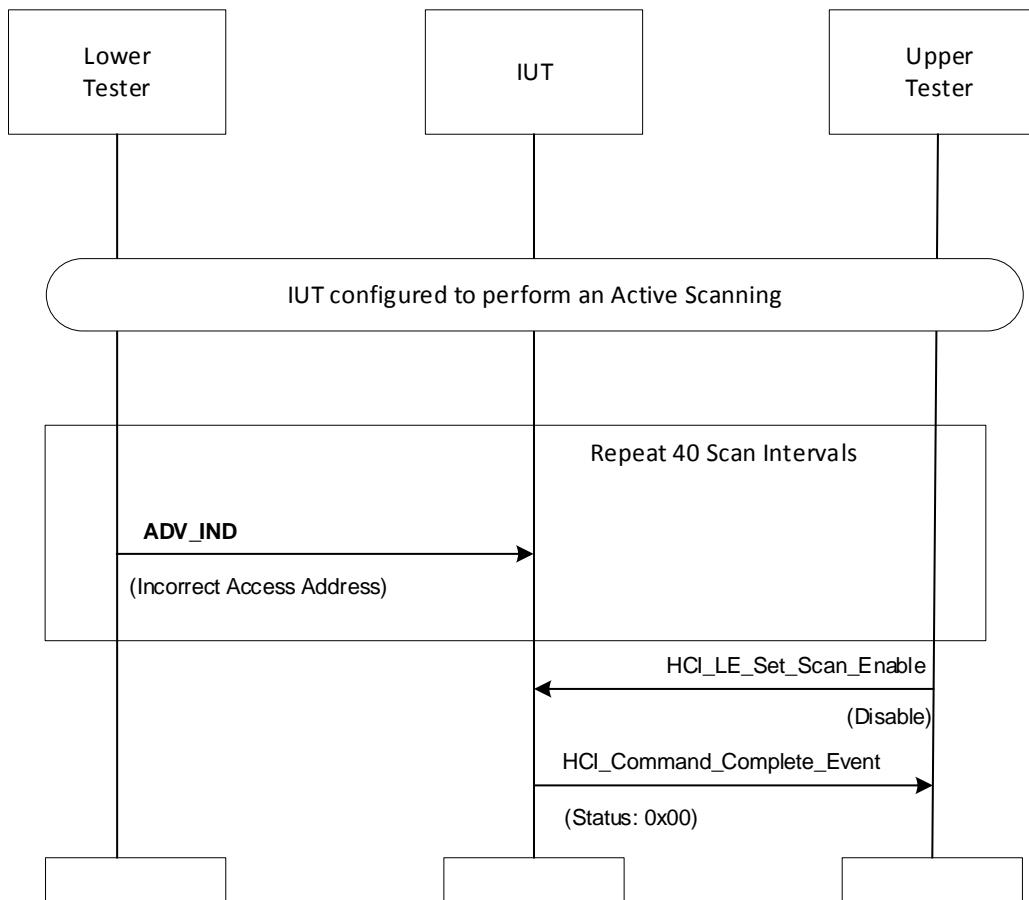


Figure 4.332: LL/ENC/SCN/BI-02-C [Active Scanning Invalid Address]

1. Upper Tester enables active scanning in the IUT.
 2. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT), undirected advertising, and a white listed device address.
 3. Lower Tester sends an ADV_IND packet with an invalid access address, each advertising event using the selected advertising channel only, using the selected advertising interval.
 4. Lower Tester expects no SCAN_REQ packet after T_IFS in response to any of the ADV_IND packets.
 5. Upper Tester expects no HCI_LE_Advertising_Report reporting the advertising packets sent by the Lower Tester.
 6. Repeat steps 3-5 until the time exceeds a number of scan intervals (40), each time changing the position of the bit which is corrupted in the access address as follows. For the first 20 ADV_IND packets sent in step 3, the Lower Tester shall corrupt bits 1 to 20 of the access address in that order. For the other 20 ADV_IND packets, it shall corrupt bits 31 to 12 of the access address in that order.
 7. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
- **Test Condition**

The parameters in this test are calculated for a BER of 0.1 % or better.

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT reporting not more than 8 of the 40 advertising indications sent in step 3.

- Notes

Bit 0 is the first bit transmitted over the air.

4.5.3 INI

Test that an initiator IUT sends a connection request to an advertiser and receiving reply transmissions with an access address different from the connection request for 6 connection intervals, considers the connection setup failed.

Note that there is no initiator test for advertising packets as the device class dependencies require that the scanner role is also implemented in a Controller with the initiator role.

4.5.3.1 LL/ENC/INI/BI-01-C [Slave Packets Invalid Address]

- Test Purpose

Test that an initiator IUT sends a connection request to an advertiser and receives reply transmissions with an access address different from the connection request for six (6) connection intervals, and considers the connection setup failed.

The Lower Tester first acts in the advertising state, accepting a connection request from the IUT, then begins to maintain the connection but with packets using a mismatching address.

- Reference

[3] 2.1.2, 4.5, 3.1

- Initial Condition

Parameters: LL_initiator_scanInterval_MIN, LL_initiator_scanInterval_MAX, LL_initiator_scanWindow_MIN, LL_initiator_scanWindow_MAX, LL_initiator_Adv_Channel_Map, LL_initiator_Channel_Map.

State: Initiating (selected scan interval, selected scan window, white list not used, public peer address, Lower Tester address, supported type of own address, common connection interval, common connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure using a selected advertising channel, advertising on the selected advertising channel only, using a selected scan interval and window using a single device address. Use the common data channel selection parameters and the common connection interval, latency and timeout to setup the connection.



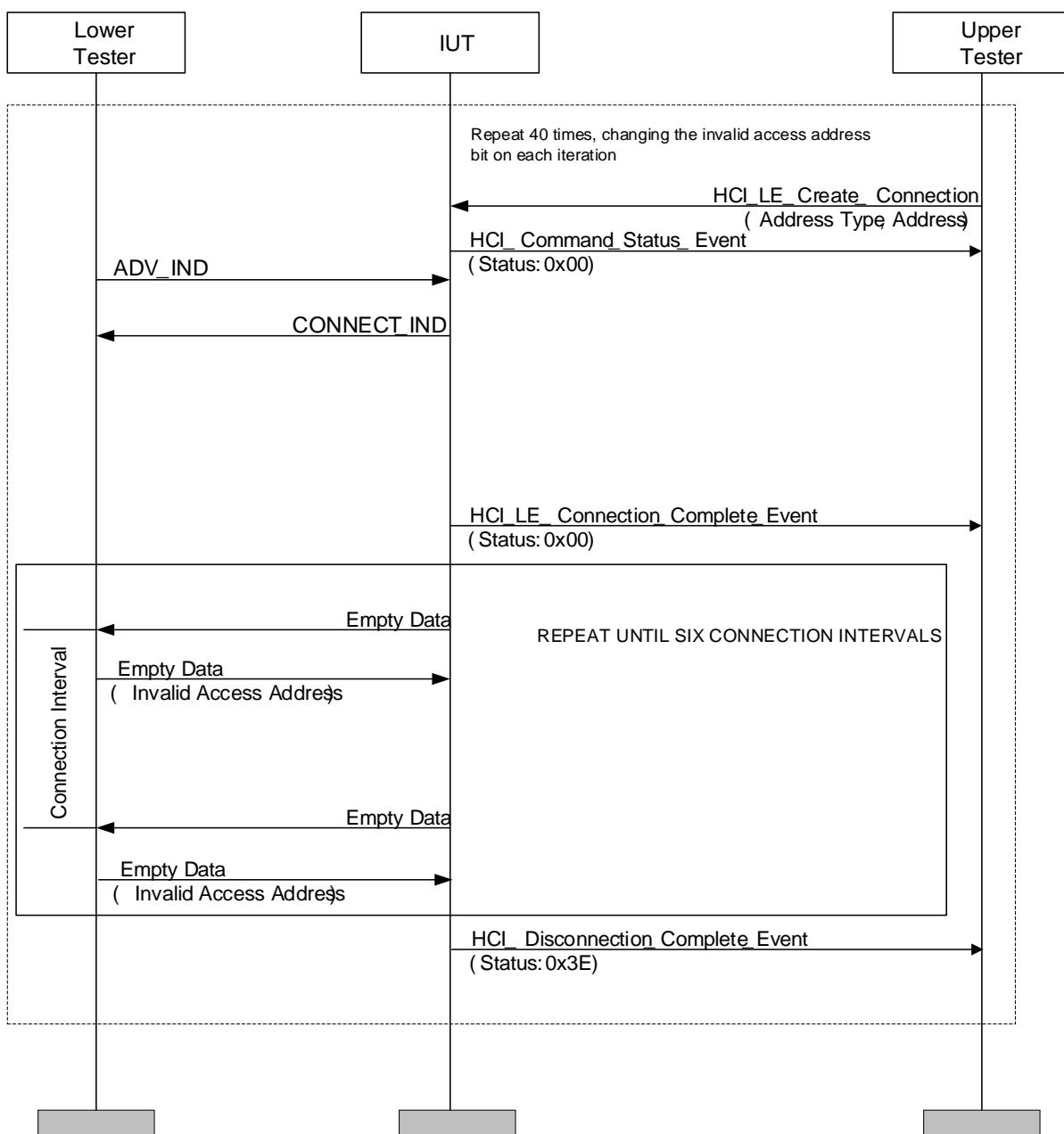


Figure 4.333: LL/ENC/IIN/BI-01-C [Slave Packets Invalid Address]

1. Configure Lower Tester as advertiser using a selected supported advertising channel (defined as an IXIT) and a common public address.
2. Upper Tester enables initiating in the IUT.
3. Lower Tester sends ADV_IND packets and expects a CONNECT_IND packet T_IFS after any of the ADV_IND packets.
4. Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the Lower Tester address and connection interval selected.
5. After the CONNECT_IND packet, expect the first empty DATA packet from the IUT on the data channel in the range of maximum/minimum deviation of the allowed transmitWindowOffset and transmitWindowSize.
6. Lower Tester sends an empty DATA packet with an access address different from the one in the connection request (corrupting a single bit in the access address), using the acknowledgement scheme, to the IUT on the same data channel.



7. Lower Tester expects following empty DATA packets on subsequent data channels.
 8. Repeat Steps 6-7 until the IUT stops.
 9. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating that a connection has failed to be established and with the connection handle matching to the one used in the connection.
 10. Repeat steps 1-9 40 times, with each iteration through the loop changing the bit position of the corrupted bit in the access address as follows. For the first 20 iterations, the Lower Tester shall corrupt bits 1 to 20 in that order. For the other 20 iterations, the Lower Tester shall corrupt bits 31 to 12 in that order.
- Test Condition
The parameters in this test are calculated for a BER of 0.1 % or better.
 - Expected Outcome
Pass Verdict
The test procedure completes the iterations using each access address with the IUT sending a connection request, then continuing master transmissions for six (6) connection intervals.
The IUT reports the failure to setup the connection with an HCI_Disconnection_Complete_Event with reason code 0x3E.
The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete event with reason code 0x3E is received in at least 32 of 40 connection attempts.
 - Notes
Bit 0 is the first bit transmitted over the air.

4.5.4 SLA

Tests that the IUT rejects packets with an invalid preamble or access address as a slave.

4.5.4.1 LL/ENC/SLA/BI-01-C [Packets to another Slave]

- Test Purpose
Tests that a slave IUT ignores a packet starting an event belonging to a different connection.
The Lower Tester acts in the master role in the connection, first setting up the connection with valid packets then changing the access address and observes the slave packets on the data channels in use.
- Reference
[3] 2.1.2, 4.5, 3.1
- Initial Condition
Parameters: LL_slave_connSlaveLatency_MIN
State: Connected Slave (common connection interval, 0 slave latency, common timeout, common channel map, any SCA value)



- Test Procedure

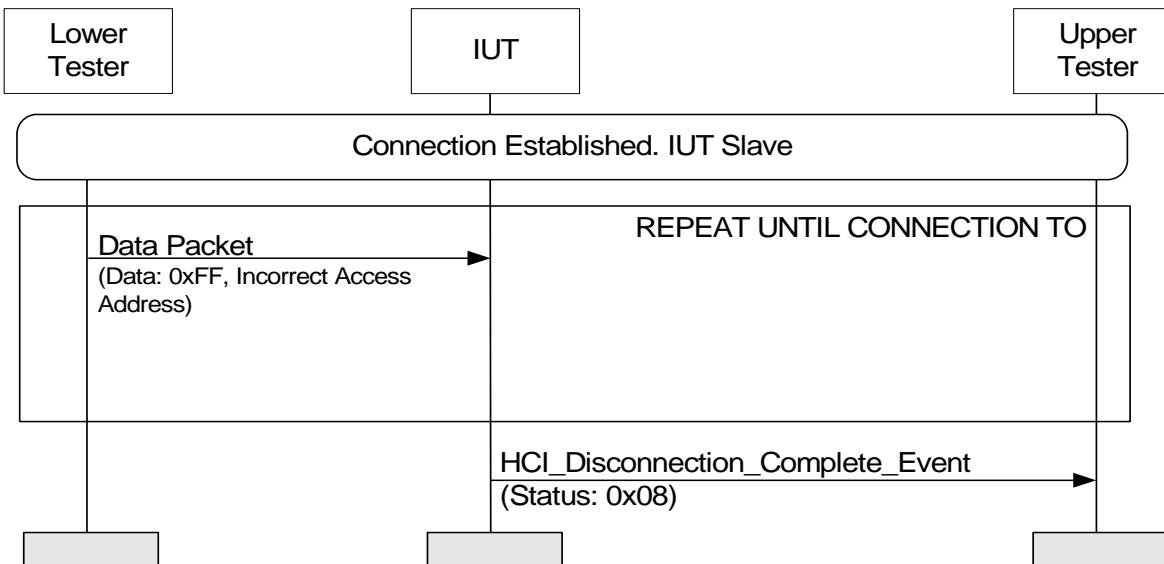


Figure 4.334: LL/ENC/SLA/BI-01-C [Packets to another Slave]

1. The Lower Tester prepares a DATA packet, corrupting the second bit in the access address.
2. The Lower Tester sends the DATA packet to the IUT using the data channel selection parameters.
3. The Lower Tester expects no DATA packet T_IFS after the packet sent.
4. The Lower Tester prepares a DATA packet, corrupting the next bit in sequence (increasing bit number) in the access address.
5. Repeat steps 2–4 until a total of 20 DATA packets have been sent in consecutive connection intervals.
6. The Lower Tester stops sending DATA packets and allows a connection supervision timeout.
7. The Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.
8. A connection is established (IUT acts as slave).
9. The Lower Tester prepares a DATA packet, corrupting the last bit in the access address.
10. The Lower Tester sends the DATA packet to the IUT using the data channel selection parameters.
11. The Lower Tester expects no DATA packet T_IFS after the packet sent.
12. The Lower Tester prepares a DATA packet, corrupting the next bit in sequence (decreasing bit number) in the access address.
13. Repeat steps 10–12 until a total of 20 DATA packets have been sent in consecutive connection intervals.
14. The Lower Tester stops sending DATA packets and allows a connection supervision timeout.
15. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.

- Expected Outcome

Pass Verdict

For each sequence of 20 DATA packets, the IUT responds in no more than 4 of the 20 connection events.

- Notes

The test allows the single bit error reverting by observing a time period longer than the connection supervision timeout. With the probability of receiving the preamble and access address correctly



around 96% and the relation of the connection interval to the connection supervision timeout value giving 10 attempts before failing, the test procedure does in practice not terminate to connection supervision timeout value under the bit error rate conditions required for an IUT not using the entire access address.

The probability of the error in the access address reverted is around 0.1%, so for 10 events received correctly and none incorrectly, the probability using binomial distribution is around 99%. Using geometric distribution to represent the number of attempts before an occurrence of 10 correctly received invalid access addresses in a row, the expected value for the repetition count is below 1 (around 0.01), so 20 events are sufficient to conclude the test procedure. This means that an IUT using the entire access address will terminate the connection.

4.5.5 MAS

Tests that the IUT rejects packets with an invalid preamble or access address procedures in the master role.

4.5.5.1 LL/ENC/MAS/BI-01-C [Packets to another Master]

- Test Purpose

Tests that a master IUT ignores packets not belonging to the connection transmitted using a different access address.

The Lower Tester acts as a slave maintaining a connection, then starts to respond with packets using an invalid access address.

- Reference

[3] 2.1.2, 4.5, 3.1

- Initial Condition

Parameters: LL_master_connSlaveLatency_MIN

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

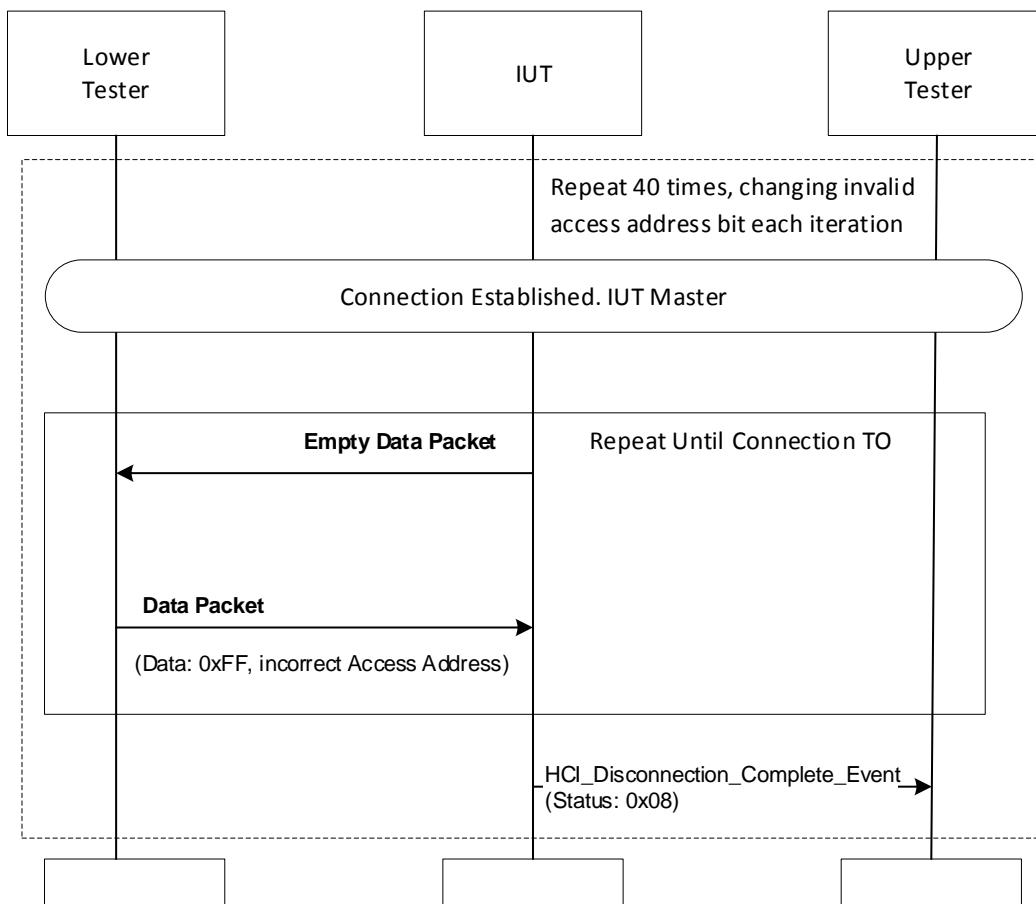


Figure 4.335: LL/ENC/MAS/BI-01-C [Packets to another Master]

1. A connection is established (IUT acts as master).
 2. Lower Tester expects a DATA packet, once a connection interval from the IUT on the data channel derived from the selection parameters.
 3. Lower Tester sends a DATA packet with an incorrect access address by corrupting a single bit in the access address.
 4. Repeat steps 1-2 for a time period equal to 2 times the connection supervision timeout value from the last valid packet sent to which an acknowledgement was received
 5. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating connection supervision timeout and containing the handle of the connection used.
 6. Repeat steps 1-5 40 times, with each iteration through the loop changing the bit position of the corrupted bit in the access address as follows. For the first 20 iterations, the Lower Tester shall corrupt bits 1 to 20 in that order. For the next 20 iterations, the Lower Tester shall corrupt bits 31 to 12 in that order.
- Expected Outcome

Pass Verdict

Each test procedure iteration executes with the parameters selected, with the connection terminated within the time period of 2 times the connection supervision timeout value from the last valid packet sent.

The acceptable success rate is 32 out of 40, i.e., that the HCI_Disconnection_Complete_Event with reason code 0x08 is received in at least 32 of 40 connections.



- Notes

Bit 0 is the first bit transmitted over the air.

4.6 FRH

Tests that the IUT behaves according to the data channel selection requirements.

4.6.1 Common PDU Contents

The packet descriptions for Lower Tester sent and Lower Tester accepted data channel packets are below.

CHANNEL_MAP_IND CtrData:

LSO MSO lsb msb	LSO MSO lsb msb
ChM (5 octets)	Instant (2 octets)

4.6.2 ADV

Tests that the IUT behaves according to the data channel selection procedure during connection setup procedure in the advertiser role.

4.6.2.1 LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]

- Test Purpose

Test that an advertiser IUT receives a connection request, stops advertising after the reception and starts to maintain a connection in the slave role using the correct data channel sequence.

The Lower Tester acts first in the initiating state, sending the connection request to the IUT, then starts to maintain a connection in the master role.

- Reference

[3] 4.5.8

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

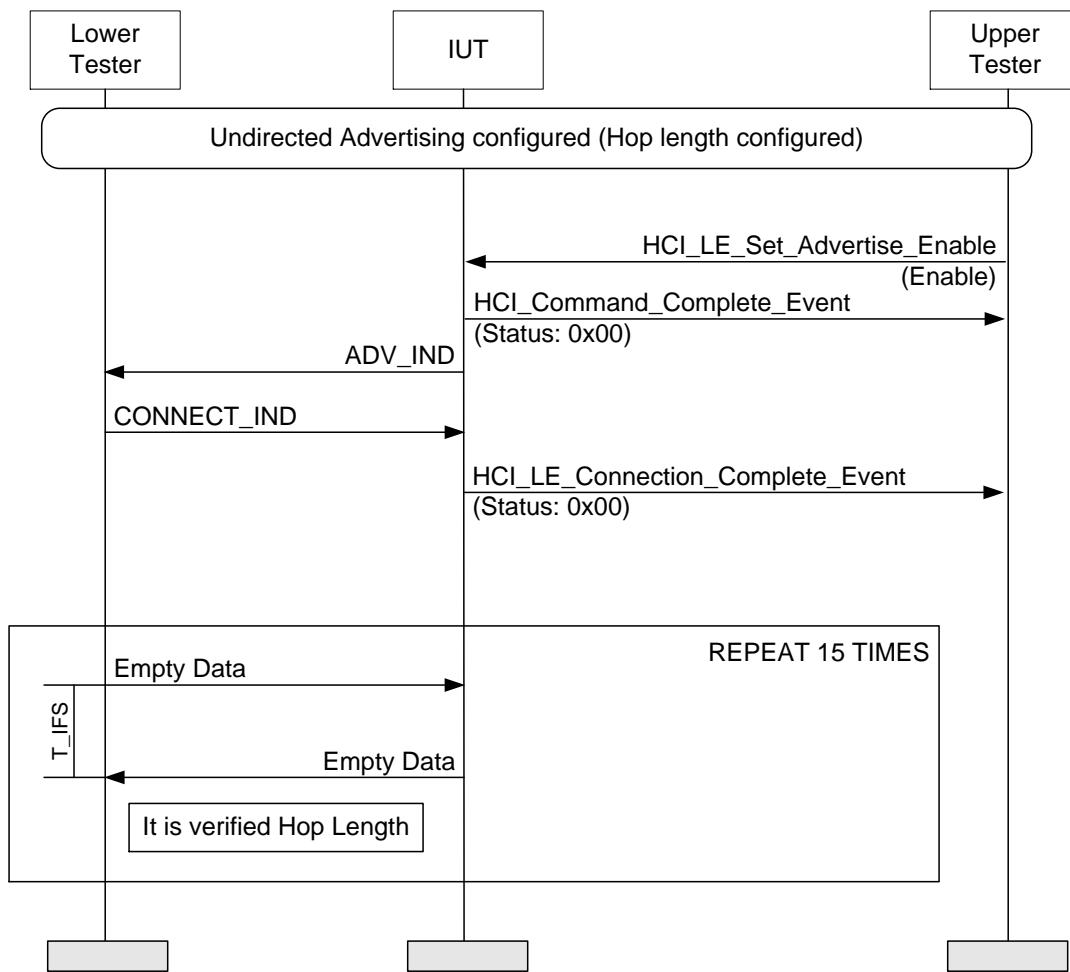


Figure 4.336: LL/FRH/ADV/BV-01-C [Accepting Connections with Hop Lengths]

- Configure Lower Tester to initiate a connection.
- Upper Tester enables advertising in the IUT using one advertising channel (defined as an IXIT).
- Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet, T_IFS after the end of the advertising packet, using a maximum hop length parameter.
- Upper Tester expects an HCI_LE_Connection_Complete event from the IUT including the parameters sent to the IUT in step 3.
- Lower Tester sends an empty LL DATA packet every connection interval, using the data channel selection parameters.
- Lower Tester expects an empty LL DATA packet from the IUT T_IFS after the data packet sent on the same data channel. Lower Tester continues sending empty data packets using the acknowledgement scheme.
- Repeat step 5-6 until receiving a response per each channel in use, or until the channel has been in use 15 times with no response.
- Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from step 4).
- Repeat steps 1-2.

10. Lower Tester expects an ADV_IND packet from the IUT on the selected advertising channel and responds with a CONNECT_IND packet using a minimum hop length parameter, TIFS after the end of the advertising packet.
 11. Repeat steps 4-8.
- Expected Outcome
- Pass Verdict
- The test procedure executes successfully with the data channel selection input variations.
- The IUT maintains the connection.
- The IUT responds on all channels at least once.

4.6.3 SLA

Tests that the IUT behaves according to the data channel selection procedure in the slave role.

4.6.3.1 LL/FRH/SLA/BV-01-C [Accepting Channel Map Update]

- Test Purpose

Test that a slave IUT accepts a channel map update request from the master and adopts the new channel map at the correct time is able to maintain the connection.

The Lower Tester acts as a master. It maintains the connection and requests a new channel map to be taken into use. It then observes the IUT accepting and adopting the channel map.
- Reference

[3] 4.5.8, 5.1.2
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value)



- Test Procedure

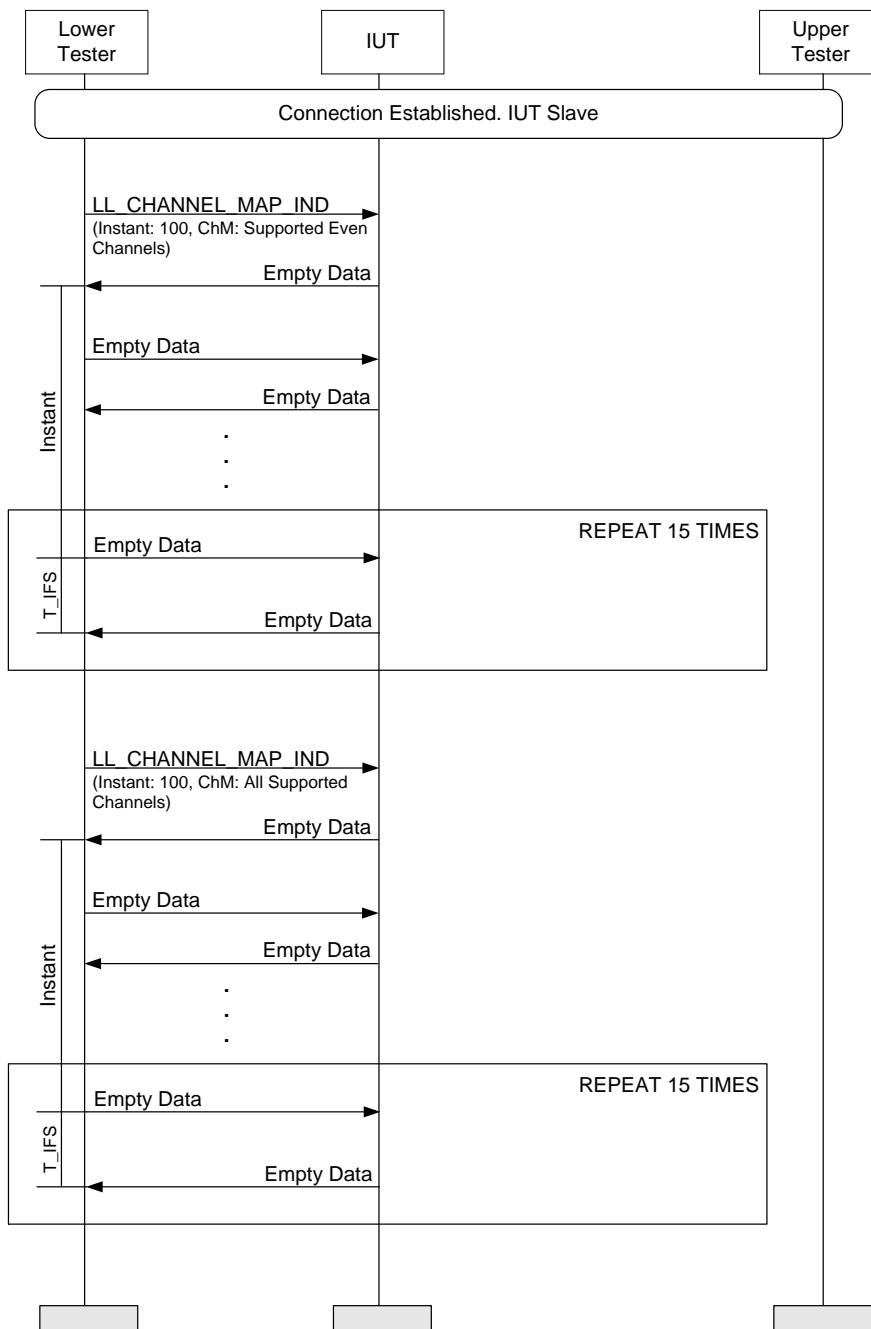


Figure 4.337: LL/FRH/SLA/BV-01-C [Accepting Channel Map Update]

- Lower Tester sends a CHANNEL_MAP_IND packet to the IUT using every second channels used as channel map and using an event count of 100. Expect an acknowledgement from the IUT.
- Lower Tester sends an empty DATA packet to the IUT and expects a response in all events until the time of the update. Repeat until the event count matches the indicated time of connection update.
- At the time of the update start maintaining the connection with the new parameters.
- Lower Tester sends an empty LL DATA packet every connection interval, using the data channel selection parameters. Expect an empty LL DATA packet from the IUT, T_IFS after the data packet sent on the same data channel. Lower Tester continues to send empty data packets using

the acknowledgement scheme. Repeat until receiving a response per each channel in use, or until the channel has been in use 15 times.

5. Lower Tester sends a CHANNEL_MAP_IND packet to the IUT using all channels used as channel map and using an event count of 100. Expect an acknowledgement from the IUT.
 6. Repeat steps 2-4
 7. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble step execution).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT acknowledging the channel map update requests and adopting the updated data channel selection parameters at the assigned event,

The IUT responds in 65 of the 100 events between the channel map request and the assigned event to adopt the updated data channel selection parameters,

- Notes

The channels applied in this test are not parameterized, assuming that the test inputs will select any channels not used.

4.6.3.2 LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2]

- Test Purpose

Tests that a slave IUT accepts a channel map update request from the master while using Channel Selection Algorithm #2, adopts the new channel map at the correct time, and maintains the connection.

The Lower Tester acts as master. It maintains the connection, requests a new channel map to be taken into use, and observes the IUT accepting and adopting the new channel map.

- Reference

[10] 4.5.8, 5.1.2

- Initial Condition

State: Connected Slave (Channel Selection Algorithm #2, any advertising interval, public address, any advertising channel map, common connection interval, common slave latency, common timeout, selected channel map, any SCA value)



- Test Procedure

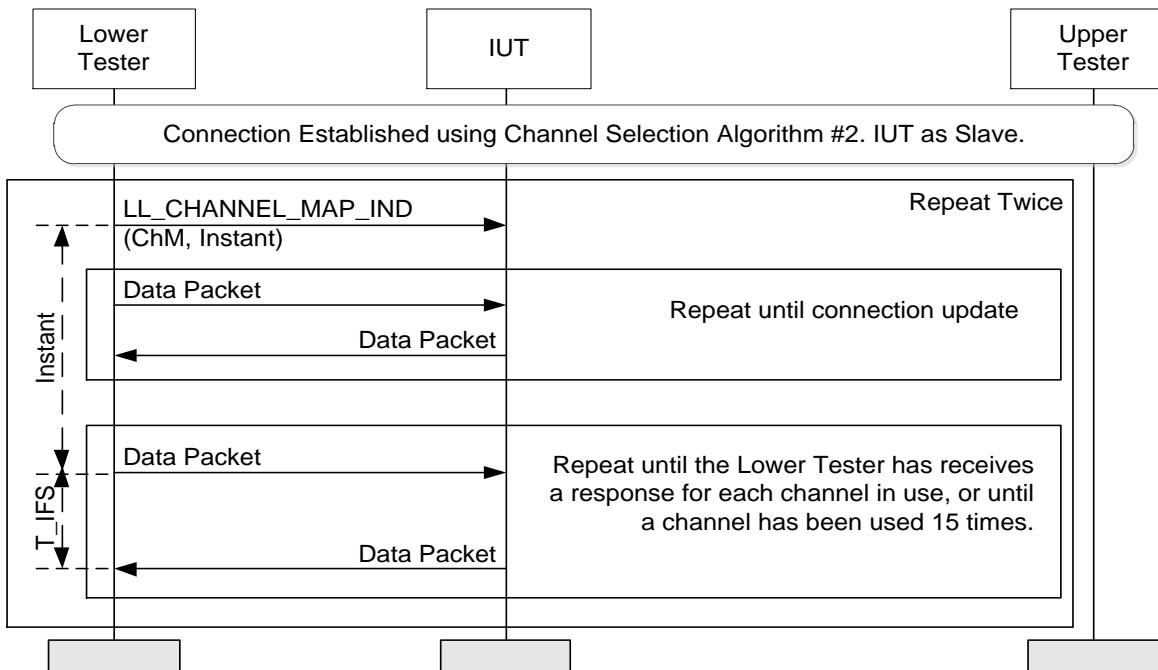


Figure 4.338: LL/FRH/SLA/BV-02-C [Accepting Channel Map Update, Channel Selection Algorithm #2]

1. The Lower Tester sends a CHANNEL_MAP_IND packet to the IUT, with the ChM field set to use every second channel (every other channel bit set to '1', others set to '0') and the Instant field set to 100 connection events ahead.
 2. The Lower Tester sends an empty LL DATA packet to the IUT every connection interval and expects a response using the acknowledgement scheme. Repeat until the event count matches the indicated time of connection update.
 3. Upon connection update, the Lower Tester switches to selecting data channel indices using Channel Selection Algorithm #2 with the new channel map.
 4. The Lower Tester sends an empty LL Data packet to the IUT every connection interval and expects a response using the acknowledgement scheme. Repeat until the Lower Tester has received a response for each channel in use, or until a channel has been used 15 times.
 5. Repeat steps 1–4, except in step 1, the ChM field shall be set to use all channels.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT accepting the channel map update request and maintaining the connection using Channel Selection Algorithm #2 with the new channel map.

The IUT sends and receives data each connection event using data channel indices selected by the Channel Selection Algorithm #2 using the new channel map.

4.6.4 MAS

Tests that the IUT behaves according to the data channel selection procedure in the master role.

4.6.4.1 LL/FRH/MAS/BV-01-C [Requesting Channel Map Update]

- Test Purpose

Test that a master IUT performs the channel map updating procedure.



The Lower Tester acts in the slave role, requests the IUT through the HCI to perform the updating procedure, accepts the request sent by the IUT and adopts the new channel map.

- Reference
[3] 4.5.8, 5.1.2
 - Initial Condition

Parameters: LL_master_Channel_Map

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

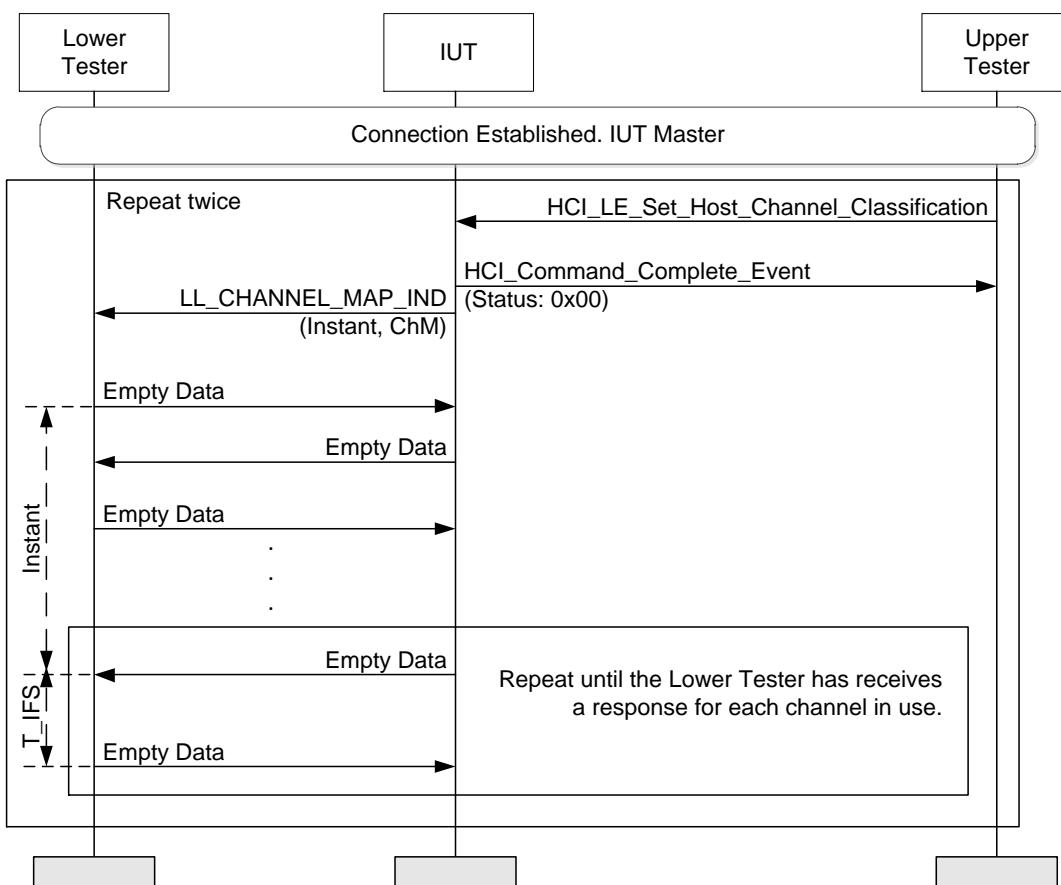


Figure 4.339: LL/FRH/MAS/BV-01-C [Requesting Channel Map Update]

1. Upper Tester sends an HCI_LE_Set_Host_Channel_Classification command to the IUT, including 'supported even channels' data channel selection parameters. Expect an HCI_Command_Complete event from the IUT in response.
 2. Lower Tester expects an LL_CHANNEL_MAP_IND control packet from the IUT, with a channel map different than the one currently in use but not necessarily matching the parameters submitted in step 1, and sends an empty DATA packet to the IUT on the same data channel using the acknowledgement scheme. If the Lower Tester does not receive the LL_CHANNEL_MAP_IND, skip steps 3-5.

3. Maintain the connection using empty DATA packets. Repeat until the event count matches the instant indicated in the channel map update request
 4. Upon the instant, take the new parameters in use.
 5. Lower Tester expects an empty LL DATA packet from the IUT and sends an empty LL DATA packet every connection interval Continue to send empty data packets using the acknowledgement scheme. Repeat until receiving a response per each channel in use. If any channel has been used 15 times without receiving a response on that channel, this test ends in a Fail Verdict.
 6. Repeat steps 1-5, except that in step 1, the Upper Tester includes 'all channels used' data channel selection parameters.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT requesting the channel map update, maintaining the connection using the original data channel selection parameters before the instant, and maintaining the connection with the updated data channel selection parameters after the instant.

Inconclusive Verdict

For both iterations, either the IUT does not send an LL_CHANNEL_MAP_IND PDU in step 2 or the map in the PDU is the same as the one currently in use.

4.6.4.2 LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2]

- Test Purpose

Tests that a master IUT performs the channel map update procedure while using Channel Selection Algorithm #2.

The Lower Tester acts in the slave role, requests the IUT through the HCI to perform the updating procedure, accepts the request sent by the IUT, and adopts the new channel map using Channel Selection Algorithm #2.

- Reference

[10] 4.5.8, 5.1.2

- Initial Condition

Parameters: LL_master_Channel_Map

State: Connected Master (Channel Selection Algorithm #2, any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

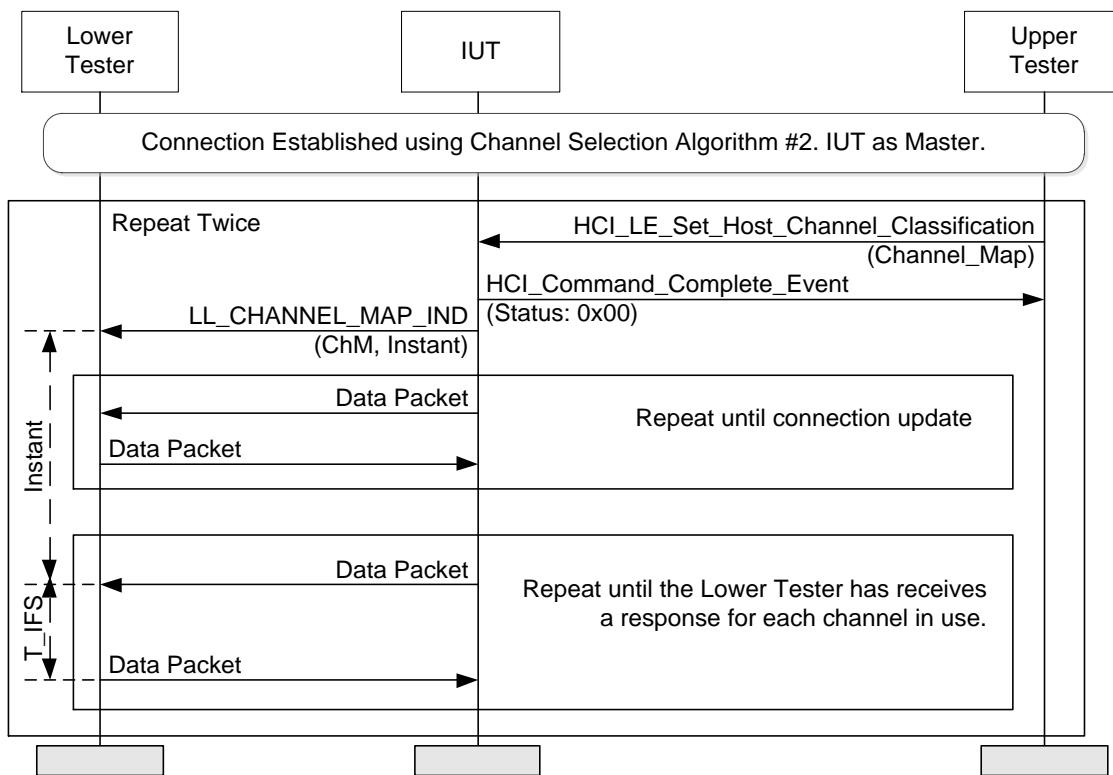


Figure 4.340: LL/FRH/MAS/BV-02-C [Requesting Channel Map Update, Channel Selection Algorithm #2]

- The Upper Tester sends an `HCI_LE_Set_Host_Channel_Classification` command to the IUT with `Channel_Map` set to mark only supported even channels as unknown (supported even channel bits set to '1', others set to '0') and expects an `HCI_Command_Complete` event in response.
- The Lower Tester expects an `LL_CHANNEL_MAP_IND` packet from the IUT, with a channel map different than the one currently in use but not necessarily matching the parameters submitted in step 1, and sends an empty LL DATA packet to the IUT on the same channel using the acknowledgement scheme. If the Lower Tester does not receive the `LL_CHANNEL_MAP_IND`, skip steps 3-5.
- The Lower Tester maintains the connection using empty LL DATA packets. Repeat until the event count matches the instant indicated in the channel map update request.
- Upon the instant, the Lower Tester switches to selecting data channel indices using Channel Selection Algorithm #2 with the new channel map.
- The Lower Tester expects LL DATA packets from the IUT and sends empty LL DATA packets every connection interval using the acknowledgement scheme. Repeat until the Lower Tester has received a response for each channel in use. If any channel has been used 15 times without receiving a response on that channel, this test ends in a Fail Verdict.
- Repeat steps 1-5, except in step 1, the `Channel_Map` shall be set to mark all supported channels as unknown (supported channel bits set to '1', others set to '0').

- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT requesting the channel map update, maintaining the connection using Channel Selection Algorithm #2 with the original data channel selection parameters before the instant, and maintaining the connection using Channel Selection Algorithm #2 with the updated data channel selection parameters after the instant.



The IUT sends and receives data each connection event using data channel indices selected by the Channel Selection Algorithm #2 using the new channel map.

Inconclusive Verdict

For both iterations, either the IUT does not send an LL_CHANNEL_MAP_IND PDU in step 2 or the map in the PDU is the same as the one currently in use.

4.6.4.3 LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels]

- Test Purpose

Tests that a master IUT performs the Minimum Number of Used Channels Procedure.

The Lower Tester acts in the slave role, sending the request to the IUT and maintaining the connection.

- Reference

[10] 5.1.11

- Initial Condition

Parameters: LL_master_Channel_Map

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

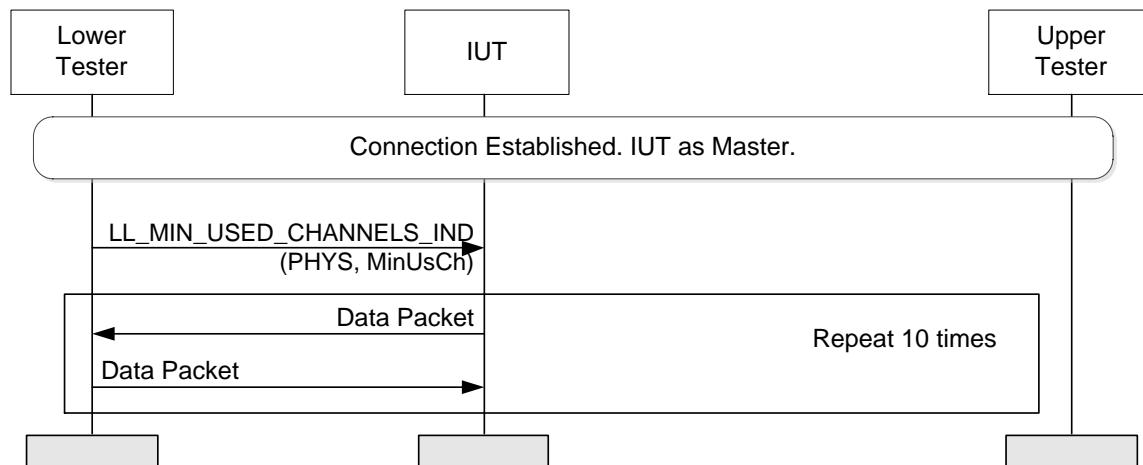


Figure 4.341: LL/FRH/MAS/BV-03-C [Accepting Minimum Number of Used Channels]

1. The Lower Tester sends an LL_MIN_USED_CHANNELS_IND PDU to the IUT. For each PHY the IUT supports, the matching bit in the PHYS field shall be set to '1'. The MinUsCh field shall be set to 0x25 (37 channels).
 2. The Lower Tester expects an LL Data packet from the IUT every connection interval and sends a response using the acknowledgement scheme. Repeat 10 times.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT receiving the LL_MIN_USED_CHANNELS_IND PDU and maintaining the connection.



4.7 PAC

Tests that the IUT rejects packets with invalid encoding contents.

4.7.1 Common PDU Contents

UNKNOWN_RSP CtrData:

LSO MSO
lsb msb
UnknownType
(1 octet)

4.7.2 SLA

Tests that the IUT rejects packets with invalid encoding contents in the slave role.

4.7.2.1 LL/PAC/SLA/BI-01-C [Control PDUs with Invalid Length from Master]

- Test Purpose

Tests that a slave IUT correctly handles invalid LL Control PDUs.

The Lower Tester sends invalid LL Control PDUs to the IUT and checks that it replies with an LL_UNKNOWN_RSP PDU each time.

- Reference

[3] 2.4.2, 2.4.2.8

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, public address, any advertising channel map, common connection interval, 0 slave latency, common timeout, common channel map, any SCA value)

- Test Procedure

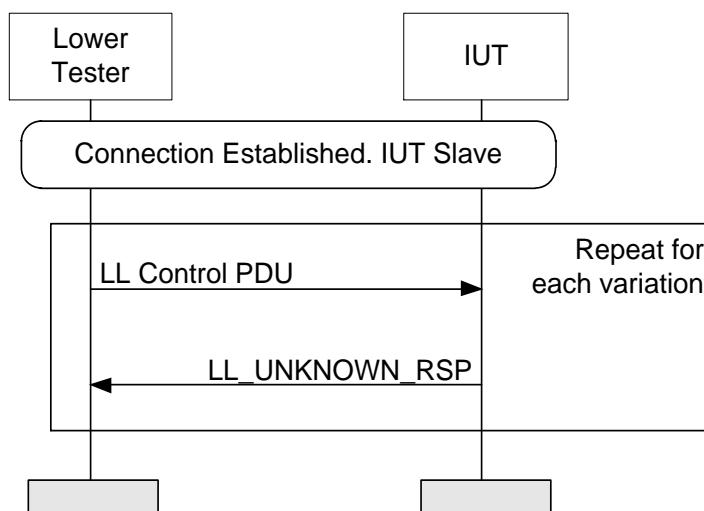


Figure 4.342: LL/PAC/SLA/BI-01-C [Control PDUs with Invalid CtrData from Master]



1. Execute steps 2-6 for each LL Control PDU opcode supported by the IUT.
 2. If the valid length of CtrData for the current opcode is greater than 0, execute steps 5-6 for CtrData length equal to 0 and for CtrData length equal to (the valid length - 1).
 3. If the valid length of CtrData for the current opcode is less than 26, execute steps 5-6 for CtrData length equal to (the valid length + 1) and for CtrData length equal to 26.
 4. Execute steps 5-6 for additional random and unique invalid CtrData lengths until 6 different invalid lengths have been used for the current opcode.
 5. The Lower Tester sends an LL Control PDU to the IUT with the current opcode and length of CtrData, using random octets for the CtrData.
 6. The Lower Tester expects to receive an LL_UNKNOWN_RSP PDU from the IUT specifying the opcode used in the LL Control PDU in step 5.
- Expected Outcome

Pass Verdict

The IUT responds to each PDU with an LL_UNKNOWN_RSP PDU specifying the opcode.

4.7.2.2 LL/PAC/SLA/BV-01-C [Unknown Packet from Master]

- Test Purpose

Test that a slave IUT responds with the unknown response packet to a device transmitting a control packet not in the supported specification or not supported by the IUT.

The Lower Tester acts in the master role and starts an event with packet including a control type indicator not used in the supported specification, and then observes the IUT response. The Lower Tester repeats this test for all opcodes that are not supported by the IUT.

- Reference

[3] 2.4.2, 2.4.2.8

- Initial Condition

Parameters: <The feature set supported by the IUT, if the specification features are not selectable for this test.>

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, common slave latency, common timeout, common channel map, any SCA value)



- Test Procedure

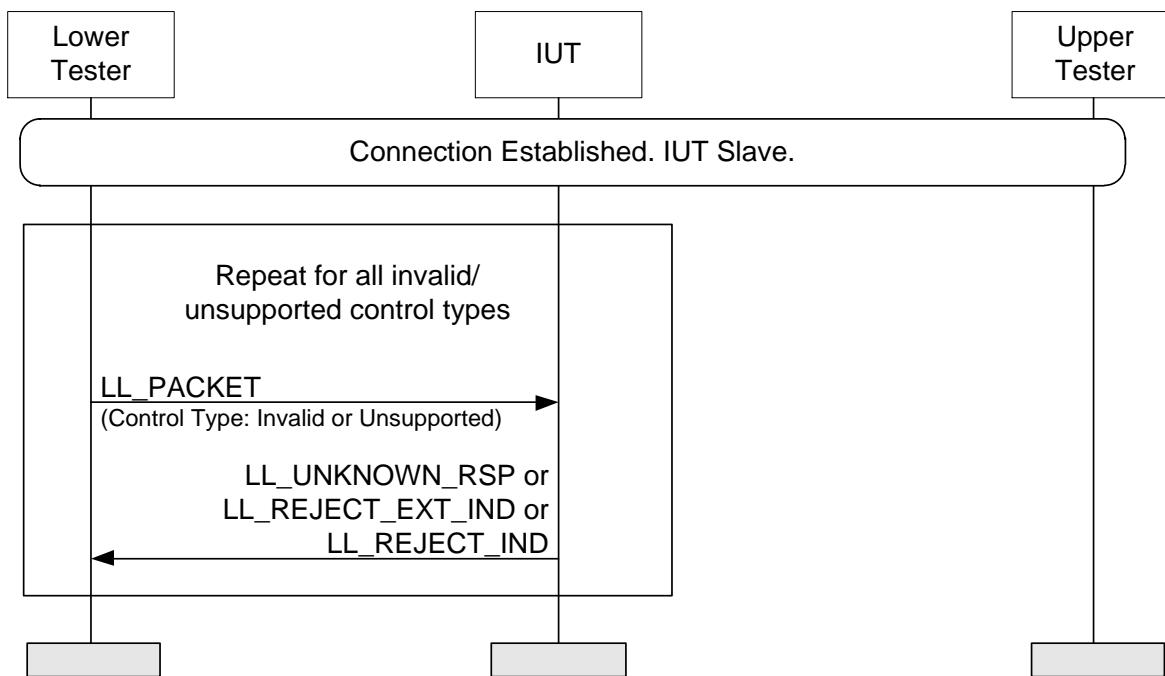


Figure 4.343: LL/PAC/SLA/BV-01-C [Unknown Packet from Master]

1. Lower Tester sends a control packet with an invalid control type value or a control type value not supported by the IUT to the IUT. This includes having the Lower Tester send opcodes that are invalid for the current role (e.g., LL_SLAVE_FEATURE_REQ sent to a slave IUT).
 2. Lower Tester expects an LL_UNKNOWN_RSP, LL_REJECT_EXT_IND, or LL_REJECT_IND PDU packet from the IUT with an error code different than success.
 3. Repeat steps 1-2 for all invalid control type values or control type values not supported by the IUT.
 4. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble step execution).
- Expected Outcome

Pass Verdict

The IUT sends an unknown response packet upon reception of all invalid/unsupported control packets.

4.7.3 MAS

Tests that the IUT rejects packets with invalid encoding contents in the master role.

4.7.3.1 LL/PAC/MAS/BI-01-C [Control PDUs with Invalid Length from Slave]

- Test Purpose

Tests that a master IUT correctly handles invalid LL Control PDUs.

The Lower Tester sends invalid LL Control PDUs to the IUT and checks that it replies with an LL_UNKNOWN_RSP PDU each time.

- Reference

[3] 2.4.2, 2.4.2.8



- Initial Condition

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

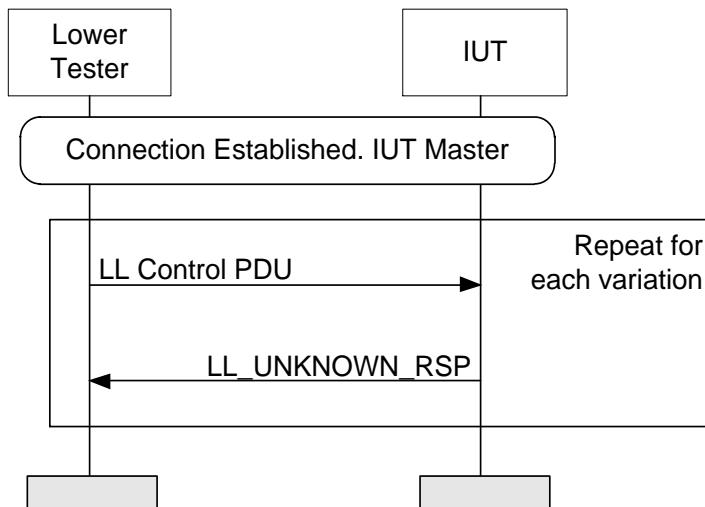


Figure 4.344: LL/PAC/MAS/BI-01-C [Control PDUs with Invalid CtrData from Slave]

- Execute steps 2-6 for each LL Control PDU opcode supported by the IUT.
- If the valid length of CtrData for the current opcode is greater than 0, execute steps 5-6 for CtrData length equal to 0 and for CtrData length equal to (the valid length - 1).
- If the valid length of CtrData for the current opcode is less than 26, execute steps 5-6 for CtrData length equal to (the valid length + 1) and for CtrData length equal to 26.
- Execute steps 5-6 for additional random and unique invalid CtrData lengths until 6 different invalid lengths have been used for the current opcode.
- The Lower Tester sends an LL Control PDU to the IUT with the current opcode and length of CtrData, using random octets for the CtrData.
- The Lower Tester expects to receive an LL_UNKNOWN_RSP PDU from the IUT specifying the opcode used in the LL Control PDU in step 5.

- Expected Outcome

Pass Verdict

The IUT responds to each PDU with an LL_UNKNOWN_RSP PDU specifying the opcode.

4.7.3.2 LL/PAC/MAS/BV-01-C [Unknown Packet from Slave]

- Test Purpose

Test whether a master IUT responds with the unknown response packet to a device transmitting a control packet not in the supported specification or not supported by the IUT.

The Lower Tester acts in the slave role, maintaining a connection, responding to the master with a packet including a control type indicator not used in the supported specification, then observes the IUT response. The Lower Tester repeats this test for all opcodes that are not supported by the IUT.

- Reference

[3] 2.4.2, 2.4.2.8



- Initial Condition

Parameters: <The feature set supported by the IUT, if the specification features are not selectable for this test.>

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout)

- Test Procedure

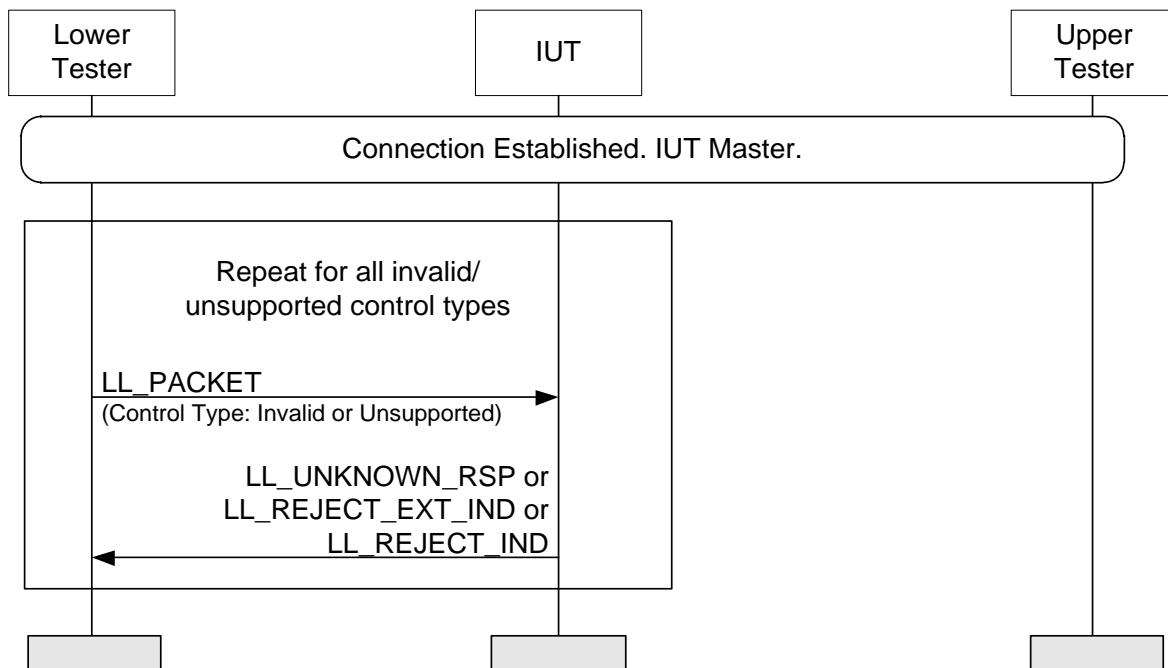


Figure 4.345: LL/PAC/MAS/BV-01-C [Unknown Packet from Slave]

- Lower Tester expects a DATA packet from the IUT and responds sending a control packet with an invalid control type value or a control type value not supported by the IUT to the IUT. This includes having the Lower Tester send opcodes that are invalid for the current role (e.g., LL_CONNECTION_UPDATE_IND or LL_CHANNEL_MAP_IND sent to a master IUT).
 - Lower Tester expects an LL_UNKNOWN_RSP, LL_REJECT_EXT_IND, or LL_REJECT_IND PDU packet from the IUT in the next event with an error code different than success.
 - Repeat steps 1-2 for all invalid control type values or control type values not supported by the IUT.
 - Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle from the preamble steps).
- Expected Outcome

Pass Verdict

The IUT sends an unknown response packet upon reception of all invalid/unsupported control packets.

4.8 SEC

Tests whether the IUT behaves according to the requirements for the security procedures. The security function specific abbreviations used are defined in [3].



4.8.1 Common PDU Contents

The packet descriptions for Lower Tester sent and Lower Tester accepted data channel packets are below.

LL_ENC_REQ CtrData:

LSO MSO lsb msb	LSO MSO lsb msb	LSO MSO lsb msb	LSO MSO lsb msb
Rand (8 octets)	EDIV (2 octets)	SKDm (8 octets)	IVm (4 octets)

LL_ENC_RSP CtrData:

LSO MSO lsb msb	LSO MSO lsb msb
SKDs (8 octets)	IVs (4 octets)

Encrypted DATA PDU:

LSO MSO lsb msb	LSO MSO lsb msb	MSO LSO msb lsb
Header (2 octets)	Payload (1-27 octets)	MIC (4 octets)

4.8.2 ADV

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

4.8.2.1 LL/SEC/ADV/BV-01-C [Advertising With Static Address]

- Test Purpose

Test that an advertiser IUT is able to advertise using a static address and performs the scan response procedure with filtering settings allowing scan requests for devices supporting filtering.

The Upper Tester generates a static address, assigns it to the IUT, and then the Lower Tester acts as an active scanner, observing the advertising packets and the static address included. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when advertising is enabled.

- Reference

[3] 4.4.2

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Generate a static address AND Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, private own address, selected advertising channels, Length of device name used, common device name) AND All White Listed (policy for advertiser).

- Test Procedure

Execute the test procedure using a static address as the advertiser address.



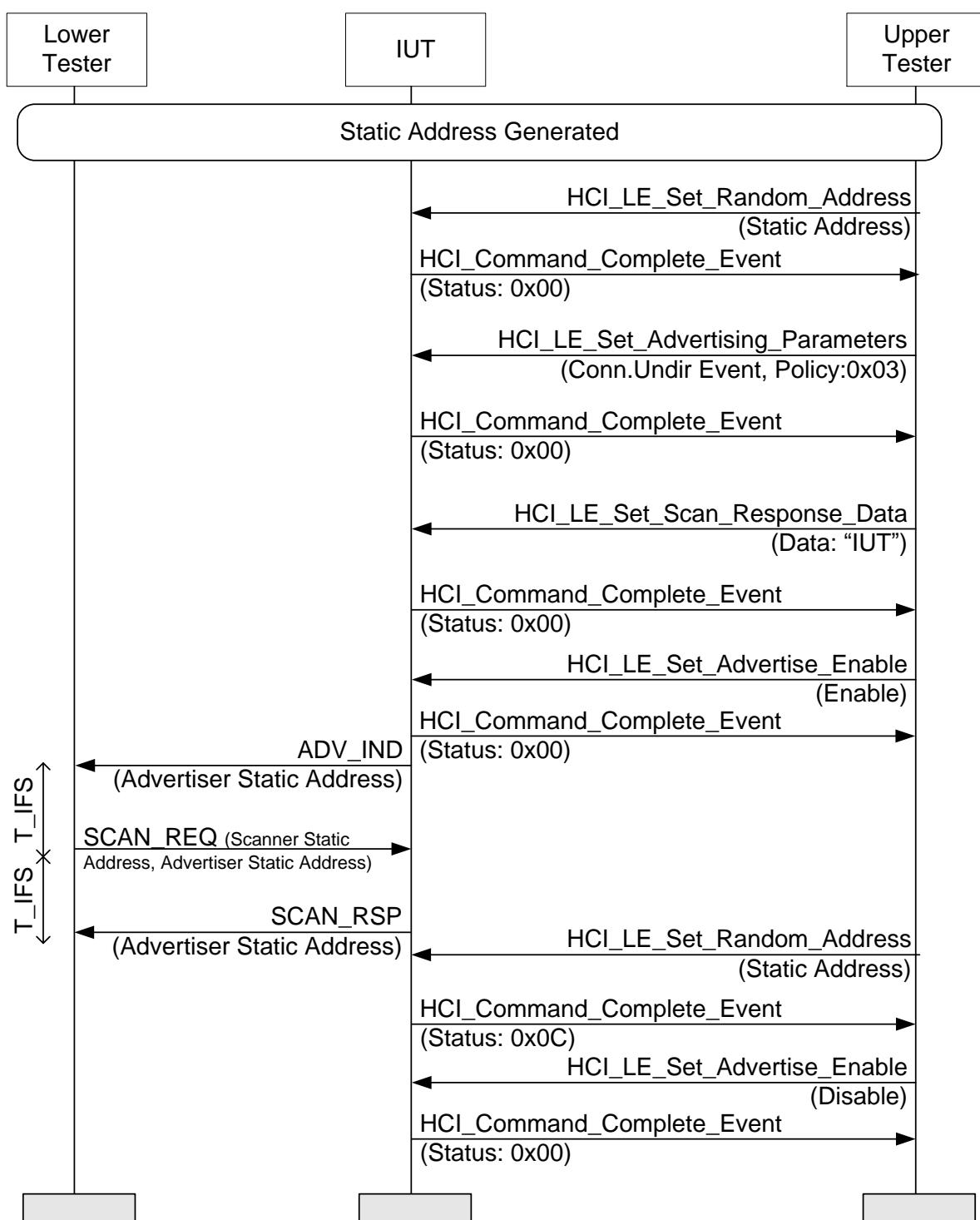


Figure 4.346: LL/SEC/ADV/BV-01-C [Advertising With Static Address]

1. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT static address.
2. Upper Tester enables undirected advertising and sets scan response data in the IUT.
3. Configure the Lower Tester to start active scanning.
4. Lower Tester expects an ADV_IND packet from the IUT using the advertising static address.
5. Lower Tester responds with a SCAN_REQ packet using the Lower Tester static address and the IUT static address on the selected advertising channel T_{IFS} after the end of an advertising packet.

6. Lower Tester expects a SCAN_RSP packet from the IUT addressed to the Lower Tester after TIFS from the end of the request packet.
 7. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT static address and expects an HCI_Command_Complete event from the IUT with a Status of 0x0C.
 8. Upper Tester sends an HCI_LE_Set_Advertise_Enable command to disable advertising in the IUT and expects an HCI_Command_Complete event from the IUT.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a static address in advertising packets.

The IUT responds to the scan requests.

The IUT rejects the second HCI_LE_Set_Random_Address command.

4.8.2.2 LL/SEC/ADV/BV-02-C [Privacy - Non-connectable Undirected Advertising, non-resolvable private address]

- Test Purpose

Verify that an advertiser IUT is able to advertise non-connectable events using a non-resolvable private address.

The Upper Tester generates a non-resolvable private address, assigns it to the IUT. The Lower Tester then observes the packet contents on the advertising channel with the non-resolvable private address included.

- Reference

[3] 4.4.2.6

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

- Test Procedure

Execute the test procedure using a single non-resolvable private address as the advertiser address and a selected advertising interval between the minimum and maximum advertising intervals supported using all supported advertising channels.



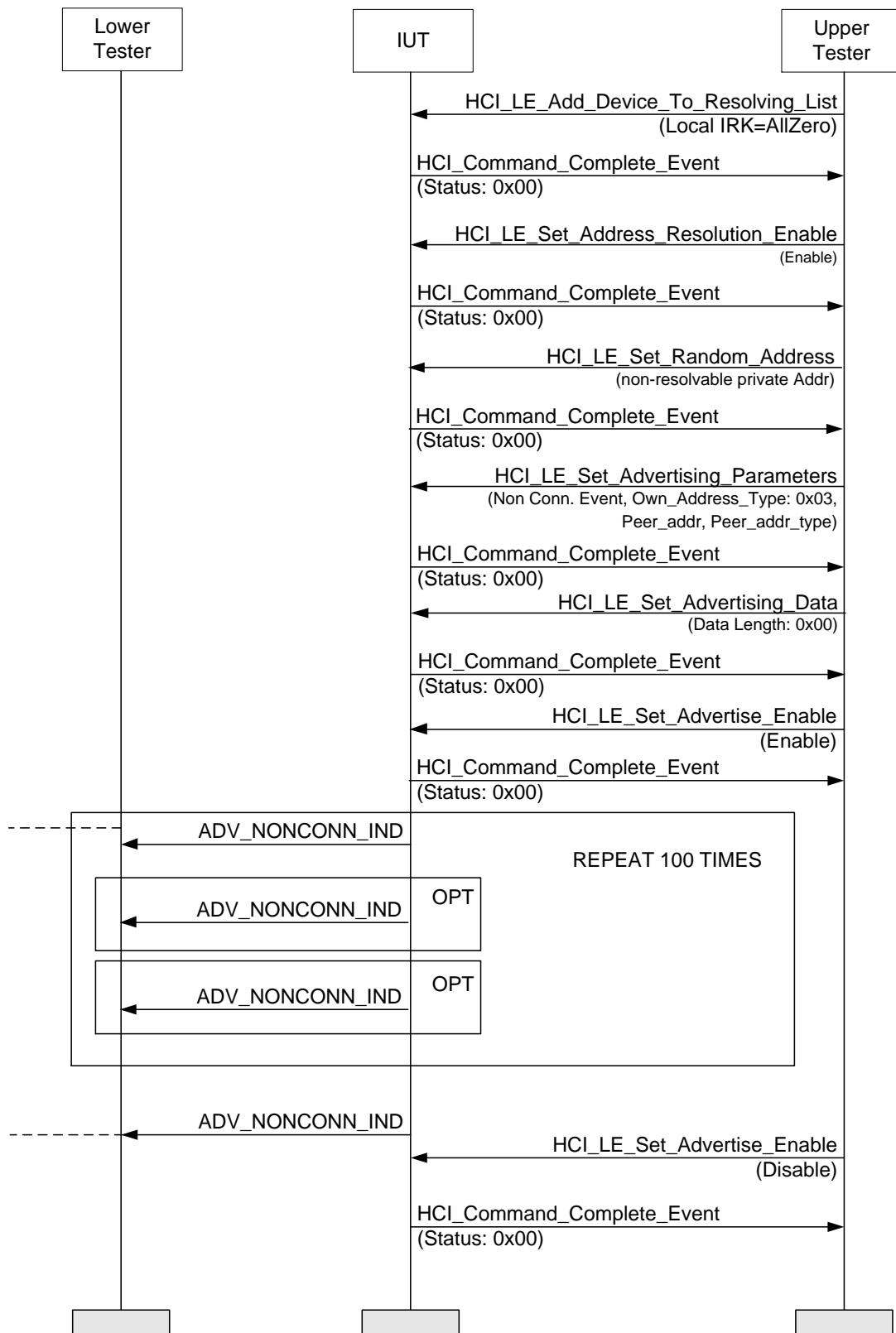


Figure 4.347: LL/SEC/ADV/BV-02-C [Privacy - Non-connectable Undirected Advertising, non-resolvable private address]

1. Configure Lower Tester to monitor advertising packets from the IUT.
 2. The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packets AdvA field.
 3. Upper Tester enables non-connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 4. Lower Tester expects the IUT to send ADV_NONCONN_IND packets on an applicable advertising channel.
 5. Repeat Steps 4 until 100 advertising events have been detected.
 6. Upper Tester disables non-connectable advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a non-resolvable private address and using it in non-connectable advertising packets.

The advertiser's address received by the Lower Tester matches the address set by the Upper Tester.

4.8.2.3 LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address]

- Test Purpose

Verify that an advertiser IUT is able to advertise non-connectable events using a resolvable private address and that the address is refreshed.

The Lower Tester observes the packet contents on the advertising channel including the resolvable private address. The resolvable private address shall be refreshed within the Resolvable Private Address timeout.

- Reference

[3] 4.4.2.6

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

State: Device Address Set (supported type of address, any address)

Execute the test procedure using a single local IRK



- Test Procedure

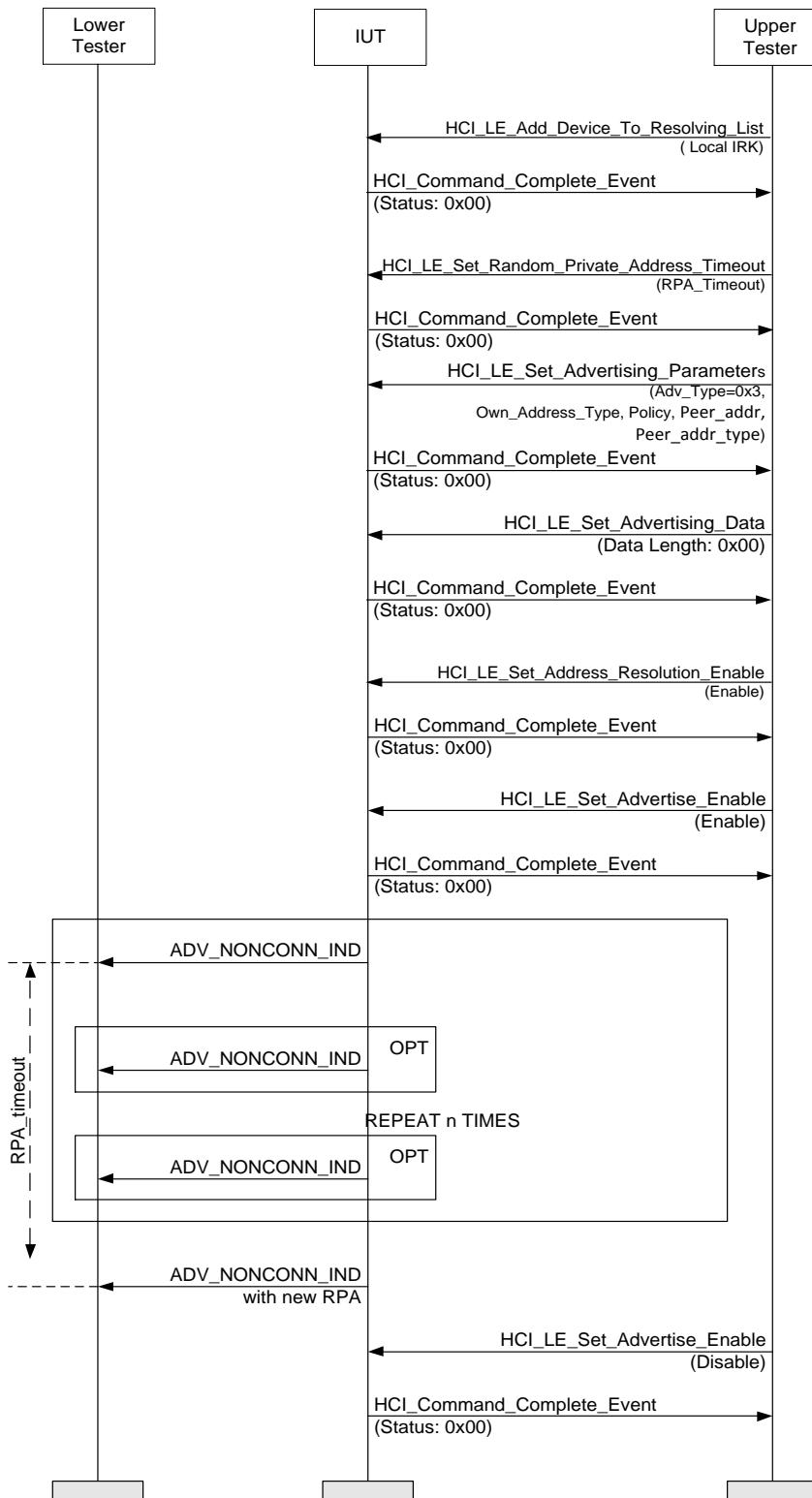


Figure 4.348: LL/SEC/ADV/BV-03-C [Privacy - Non-connectable Undirected Advertising, Resolvable Private Address]



1. Configure Lower Tester to monitor advertising packets from the IUT.
 2. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertising packet's AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.
 3. Upper Tester enables non-connectable advertising in the IUT using all supported advertising channels, a selected advertising interval between the minimum and maximum advertising intervals, and sets the Own_Address_Type to either 0x02 or 0x03 according to IUT address set in initial condition.
 4. Lower Tester expects the IUT to send ADV_NONCONN_IND packets on an applicable advertising channel.
 5. Lower Tester resolves private address received from the IUT using assigned IRK.
 6. Lower Tester waits for address refresh timeout and resolves another ADV_NONCONN_IND packet verifying that a different address is used that resolves with the same IRK.
 7. Upper Tester disables non-connectable advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in advertising and refreshing the address within the defined timeout period.

4.8.2.4 LL/SEC/ADV/BV-04-C [Network Privacy - Scannable Advertising, non-resolvable private address]

- Test Purpose

Verify that that an advertiser IUT is able to advertise scannable undirected events using a non-resolvable private address.

The Upper Tester generates a non-resolvable private address, assigns it to the IUT. The Lower Tester then observes the packet contents on the advertising channel with the non-resolvable private address included.

- Reference

[3] 4.4.2.5

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Scannable undirected advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address, selected advertising channel map, length of device name used, common device name)

Execute the test procedure using a single non-resolvable private address as the advertiser address.



- Test Procedure

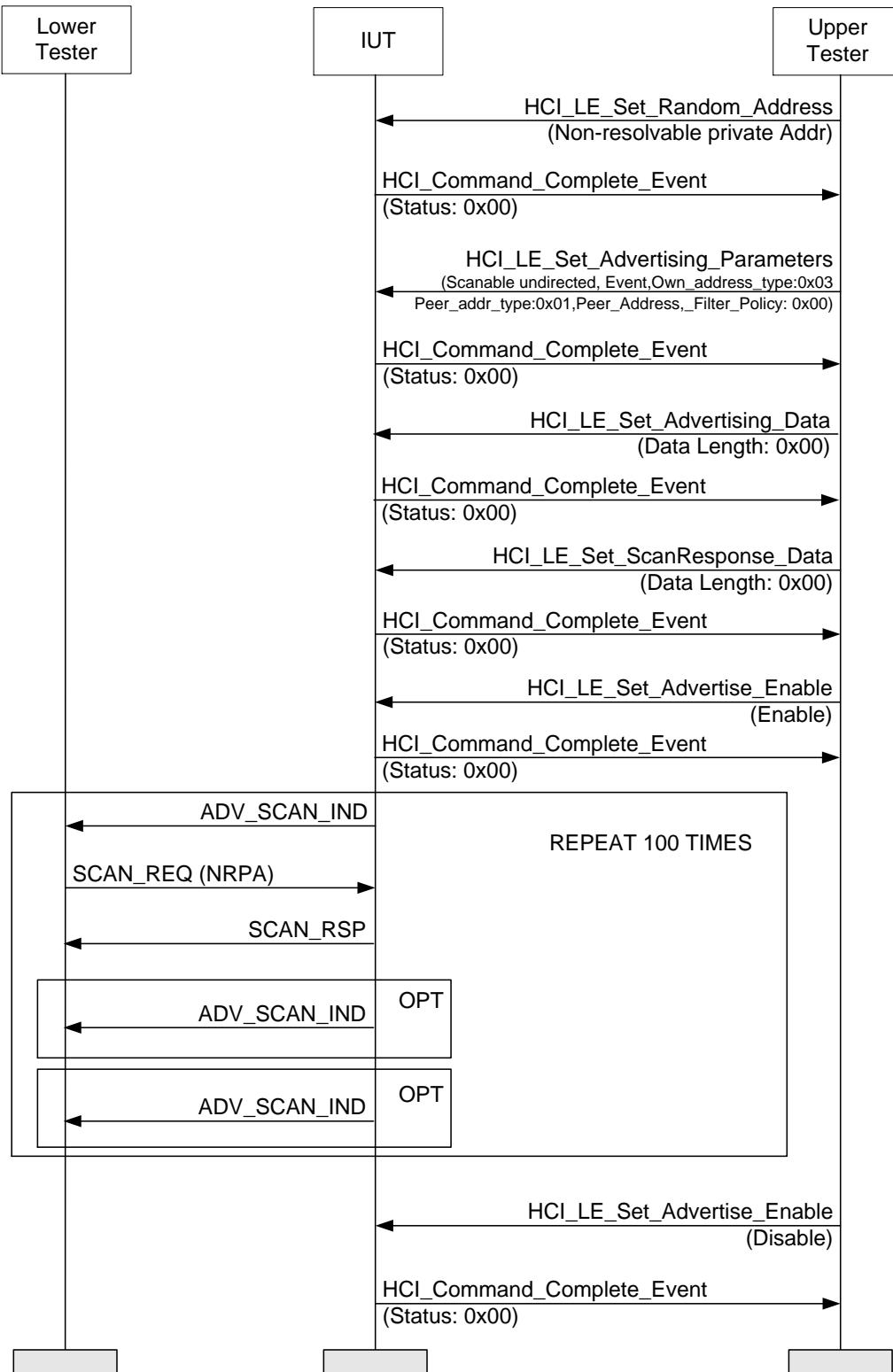


Figure 4.349: LL/SEC/ADV/BV-04-C [Network Privacy - Scannable Advertising, non-resolvable private address]

1. Configure Lower Tester to active scan for advertising packets from the IUT.
 2. Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 3. The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packet's AdvA field.
 4. Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT. The Lower Tester uses a non-resolvable private address in the SCAN_REQ packet.
 5. Lower Tester expects the IUT to send SCAN_RSP packets in response to the SCAN_REQ.
 6. Repeat Steps 4–5 until 100 advertising events have been detected.
 7. Upper Tester disables scannable undirected advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a non-resolvable private address and using it in the AdvA field of scannable undirected advertising packets.

The advertiser address matches the address received by the Lower Tester.

4.8.2.5 LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address]

- Test Purpose

Verify that an advertiser IUT is able to advertise scannable undirected events using a resolvable private address. The resolvable private address shall be refreshed within the Resolvable Private Address timeout

The Lower Tester observes the packet contents on the advertising channel, and resolves the IUT's resolvable private address. The Lower Tester then responds with a scan request using a resolvable private address and expects a scan response packet in return.

- Reference
- [3] 4.4.2.5
- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

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

Execute the test procedure using a single local IRK.



- Test Procedure

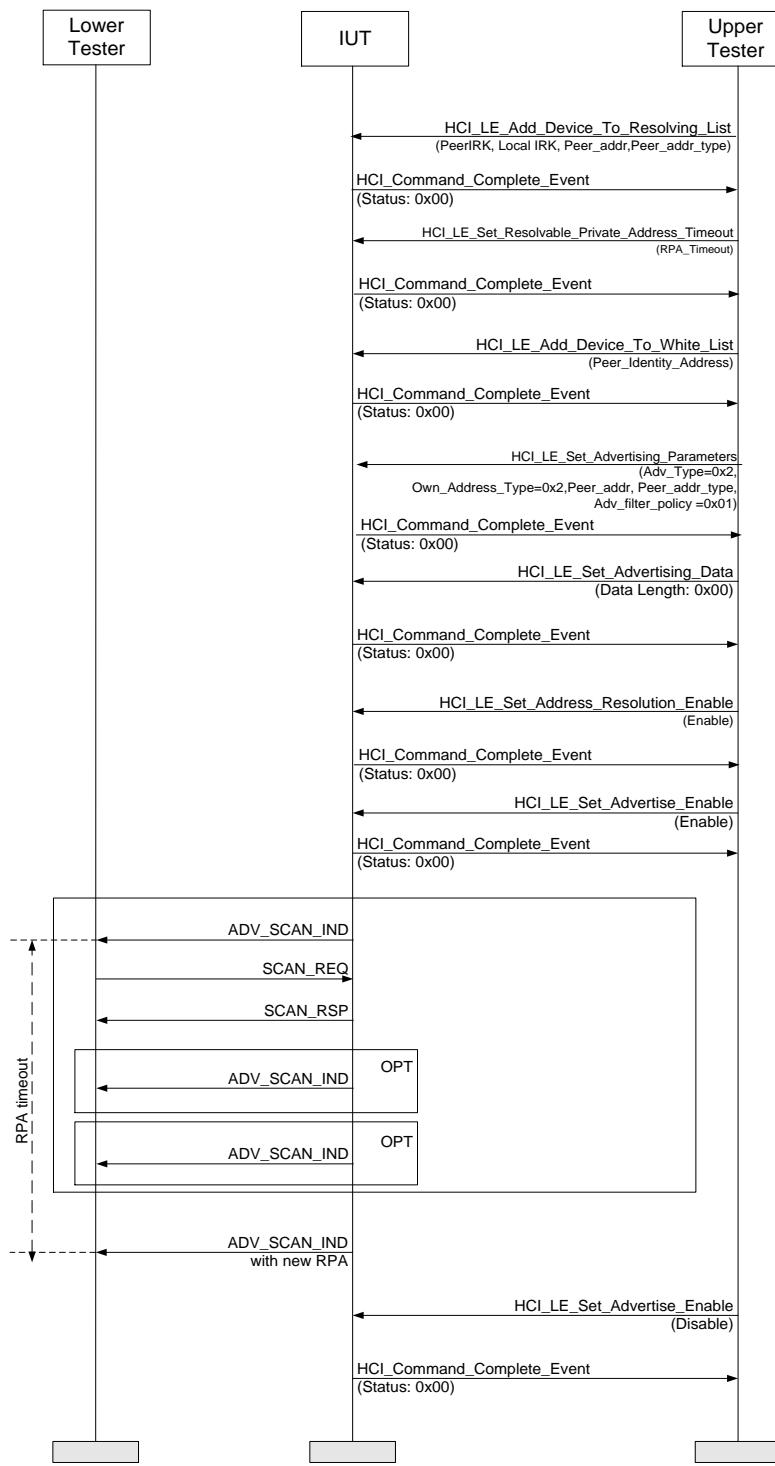


Figure 4.350: LL/SEC/ADV/BV-05-C [Network Privacy - Scannable Advertising, resolvable private address]

1. Configure Lower Tester to monitor advertising packets from the IUT.
 2. Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 3. Upper Tester adds the Lower Tester's Identity Address to the White List

4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertising packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.
 5. Upper Tester enables scannable undirected advertising.
 6. Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
 7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT.
 8. IUT resolves the private address received from the Lower Tester using assigned IRK. Lower Tester expects the IUT to send SCAN_RSP packets in response to the SCAN_REQ.
 9. Lower Tester waits for address refresh timeout and resolves another ADV_SCAN_IND packet verifying that a different address is used that resolves with the same IRK.
 10. Upper Tester disables non-connectable advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising and refreshing the address within the defined timeout period.

The Lower Tester and IUT successfully resolved the peer's address and send the SCAN_REQ and SCAN_RSP respectively.

4.8.2.6 LL/SEC/ADV/BV-06-C [Network Privacy - Undirected Connectable Advertising no Local IRK, no peer IRK]

- Test Purpose

Verify that an advertiser IUT can connect while using non-resolvable private address in the AdvA field.

- Reference

[3] 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

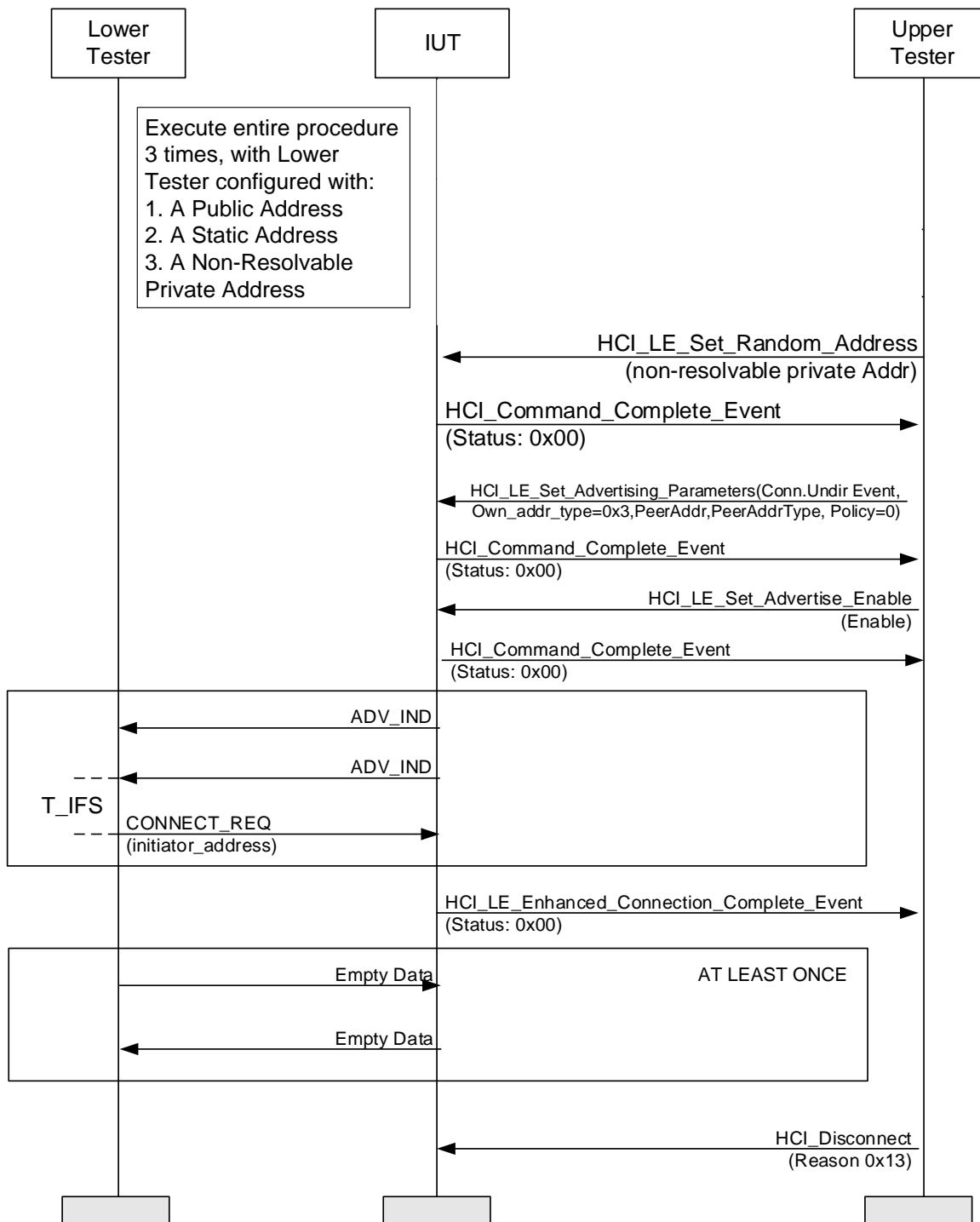


Figure 4.351: LL/SEC/ADV/BV-06-C [Network Privacy - Undirected Connectable Advertising no Local IRK, no peer IRK]

- Configure Lower Tester to initiate a connection while using a Public Address.
- The Upper Tester assigns a non-resolvable private address to the IUT for use in the advertising packets AdvA field.
- Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.



4. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
 5. Lower Tester connects to the IUT. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 6. Upper Tester terminates the connection.
 7. Repeat steps 1-6 an additional 2 times, but configure the Lower Tester with a Static Address and Non-Resolvable Private Address, respectively.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a non-resolvable private address in connectable advertising, connecting to the Lower Tester.

4.8.2.7 LL/SEC/ADV/BV-07-C [Network Privacy - Undirected Connectable Advertising with Local IRK, no peer IRK]

- Test Purpose

Verify that an advertiser IUT can connect while using the Resolving List and using a resolvable private address in the AdvA field. The peer has not distributed its IRK. The Device Identity (IRK and Identity Address) of the IUT is known by the peer.

The Lower Tester has not distributed its own IRK and uses a Public or Static Address (Identity Address) for the InitA field, the identity address is stored in the IUT's White List.

- Reference
[3] 4.4.2.3
- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

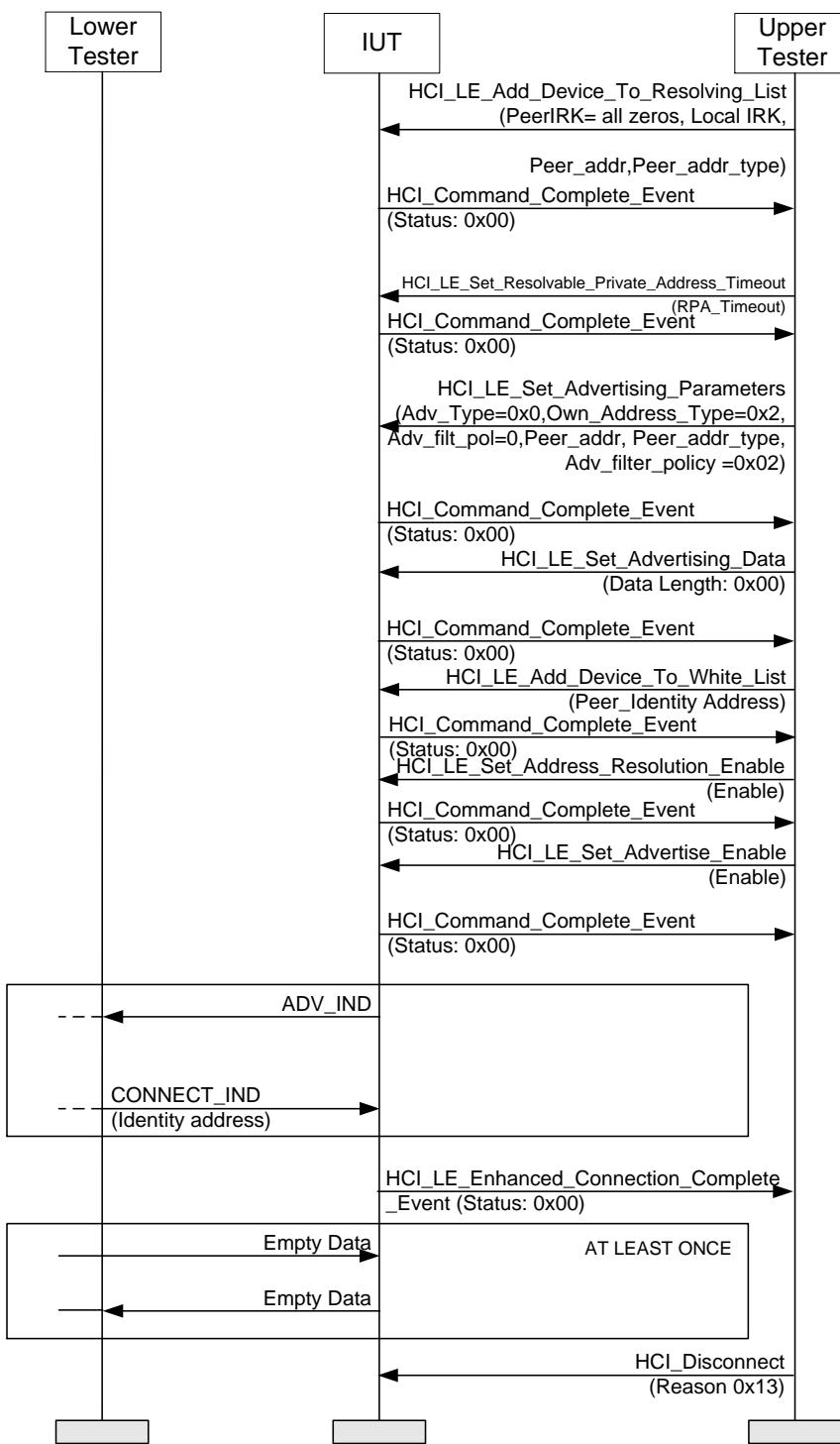


Figure 4.352: LL/SEC/ADV/BV-07-C [Network Privacy - Undirected Connectable Advertising with Local IRK, no peer IRK]

- Configure Lower Tester to initiate a connection while using the Identity Address.
- Upper Tester adds the Lower Tester's Identity Address to the White List



3. The Upper Tester adds the Lower Tester's Device Identity (IRK and Identity Address) to the IUT resolving list.
 4. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 5. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
 6. Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA identical to its Identity Address.
 7. The IUT verifies the AdvA and the InitA address of the CONNECT_IND. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

The IUT connects to the Lower Tester when the Lower Tester uses either a public or a static address.

4.8.2.8 LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK]

- Test Purpose

Verify the IUT when transmitting undirected connectable advertising events, using the Resolving List and using a resolvable private address for AdvA field, connects to the Lower Tester. The Lower Tester uses a resolvable private address for the InitA field, i.e. the Lower Tester has distributed its own device identity (IRK and Identity Address) and uses resolvable private addresses. IUT and Lower Tester validate the address used towards the device identities.

- Reference
- [3] 4.4.2.3
- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

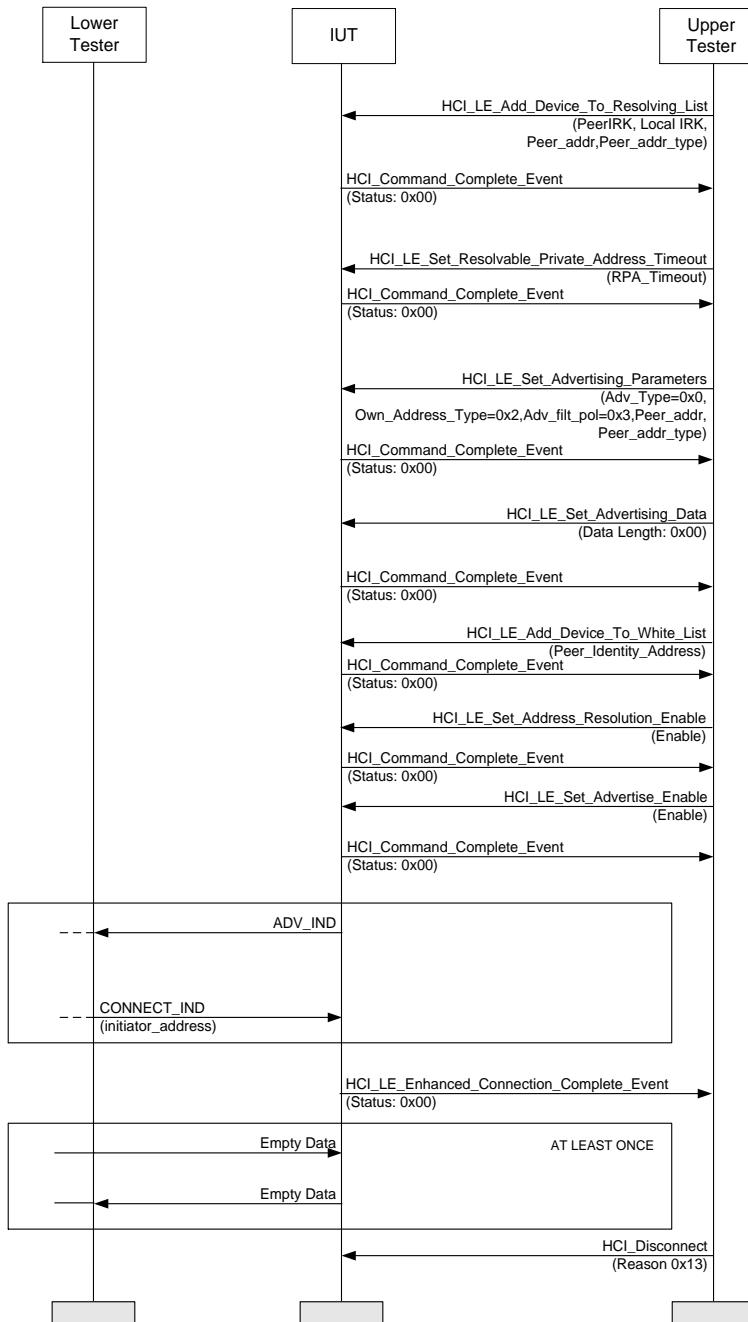


Figure 4.353: LL/SEC/ADV/BV-08-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK]

- Configure Lower Tester to initiate a connection while using a resolvable private address.
- The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK. The IUT uses this when generating a resolvable private address for use in the advertising packet's AdvA field.
- Upper Tester adds the Lower Tester's Identity Address to the White List
- Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
- Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.



6. Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its IRK. The IUT verifies the AdvA field, and resolves the address in the InitA field.
 7. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

The Lower Tester also uses a resolvable private address during the connections and the IUT correctly resolves the address.

4.8.2.9 LL/SEC/ADV/BV-09-C [Network Privacy - Undirected Connectable Advertising without Local IRK and with peer IRK]

- Test Purpose

Verify that the IUT connects to the Lower Tester when transmitting undirected connectable advertising events and using the Resolving List with a public or random static address for AdvA field. The Lower Tester has distributed its Device Identity (IRK and Identity Address) and uses a resolvable private address for the InitA field.

- Reference

[3] 4.4.2.3

- Initial Condition

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX, LL_advertiser_Adv_Channel_Map

State: Undirected Advertising (selected Adv_Interval_Min, selected Adv_Interval_Max, supported type of own address (0x02 or 0x03), selected advertising channel map)

The Upper Tester host has not provided an IRK for the IUT.



- Test Procedure

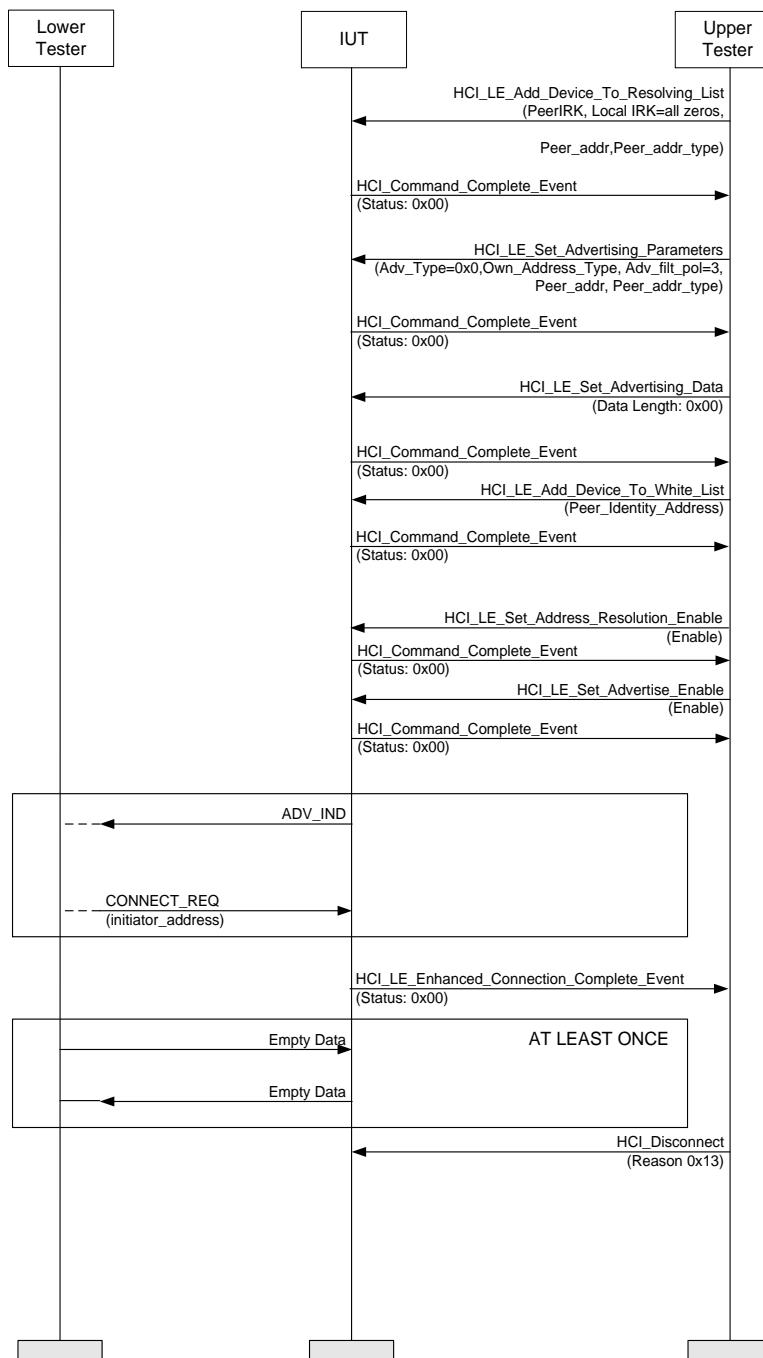


Figure 4.354: LL/SEC/ADV/BV-09-C [Network Privacy - Undirected Connectable Advertising without Local IRK and with peer IRK]

- Configure Lower Tester to initiate a connection while using a resolvable private address.
- Upper Tester adds the Lower Tester's Identity Address to the White List
- The Upper Tester populates the resolving list with the device identity of the Lower Tester and a local IRK value of all zeros.
- Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The AdVA value is set to the IUT's Identity Address.
- Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.



6. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its Device Identity. The IUT resolves the InitA Address and identifies the Lower Tester.
 7. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

IUT advertises with a Public or Static address in the AdvA field.

The IUT is able to use the resolving list to connect to the Lower Tester

4.8.2.10 LL/SEC/ADV/BV-10-C [Network Privacy - Undirected Connectable Advertising using Resolving List and Peer Device Identity not in the List]

- Test Purpose

Verify that the IUT, when transmitting undirected connectable advertising events and using the Resolving List, connects to the devices that are only resolved and on the White List. The IUT should only connect to the Lower Tester upon successful resolution of the peer's resolvable private address.

- Reference

[3] 4.4.2.3

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

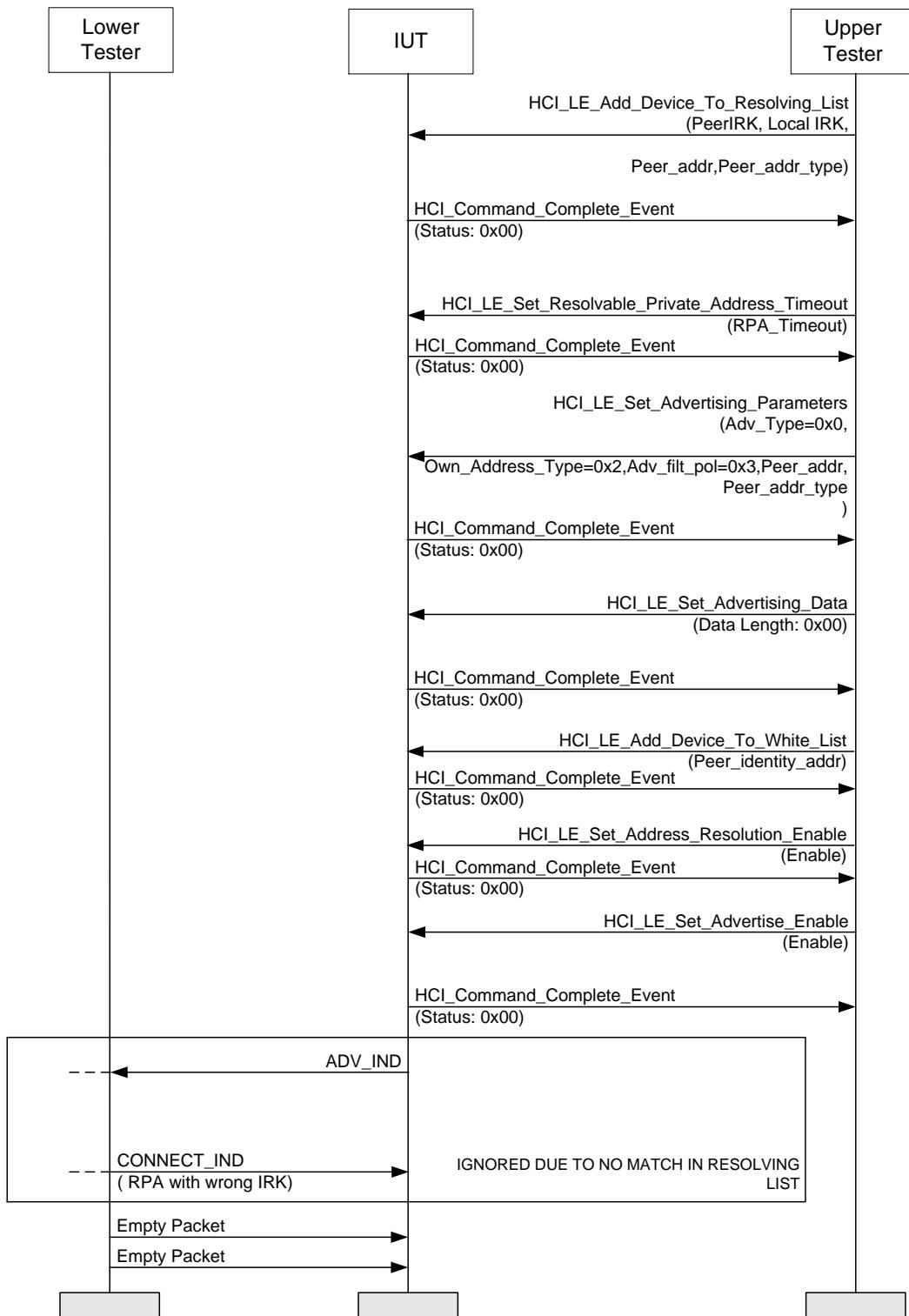


Figure 4.355: LL/SEC/ADV/BV-10-C [Network Privacy - Undirected Connectable Advertising using Resolving List and Peer Device Identity not in the List]

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Upper Tester adds the Lower Tester's Identity Address to the White List.
3. Configure the Lower Tester to initiate a connection while using a resolvable private address.



4. The Upper Tester populates the resolving list with device identities not equal to the one of the Lower Tester connected with the local device identity.
 5. The IUT use its device identity when generating a resolvable private address for use in the advertising packet's AdvA field.
 6. Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 7. Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
 8. Lower Tester sends a CONNECT_IND using the advertiser's address for AdvA field. The InitA field uses a resolvable private address based on the device identity of the Lower Tester.
 9. The Lower Tester repeats step 7 ten times.
- Expected Outcome

Pass Verdict

IUT advertises with a resolvable private address in the AdvA field.

The IUT ignores the connect requests from addresses which are not in the resolving list.

4.8.2.11 LL/SEC/ADV/BV-11-C [Network Privacy - Directed Connectable Advertising using local and remote IRK]

- Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, is using resolvable private addresses for AdvA and InitA fields when the Lower Tester has distributed its own IRK.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

Lower Tester is using a resolvable private address and has previously distributed its IRK.

The IUT is using a resolvable private address.



- Test Procedure

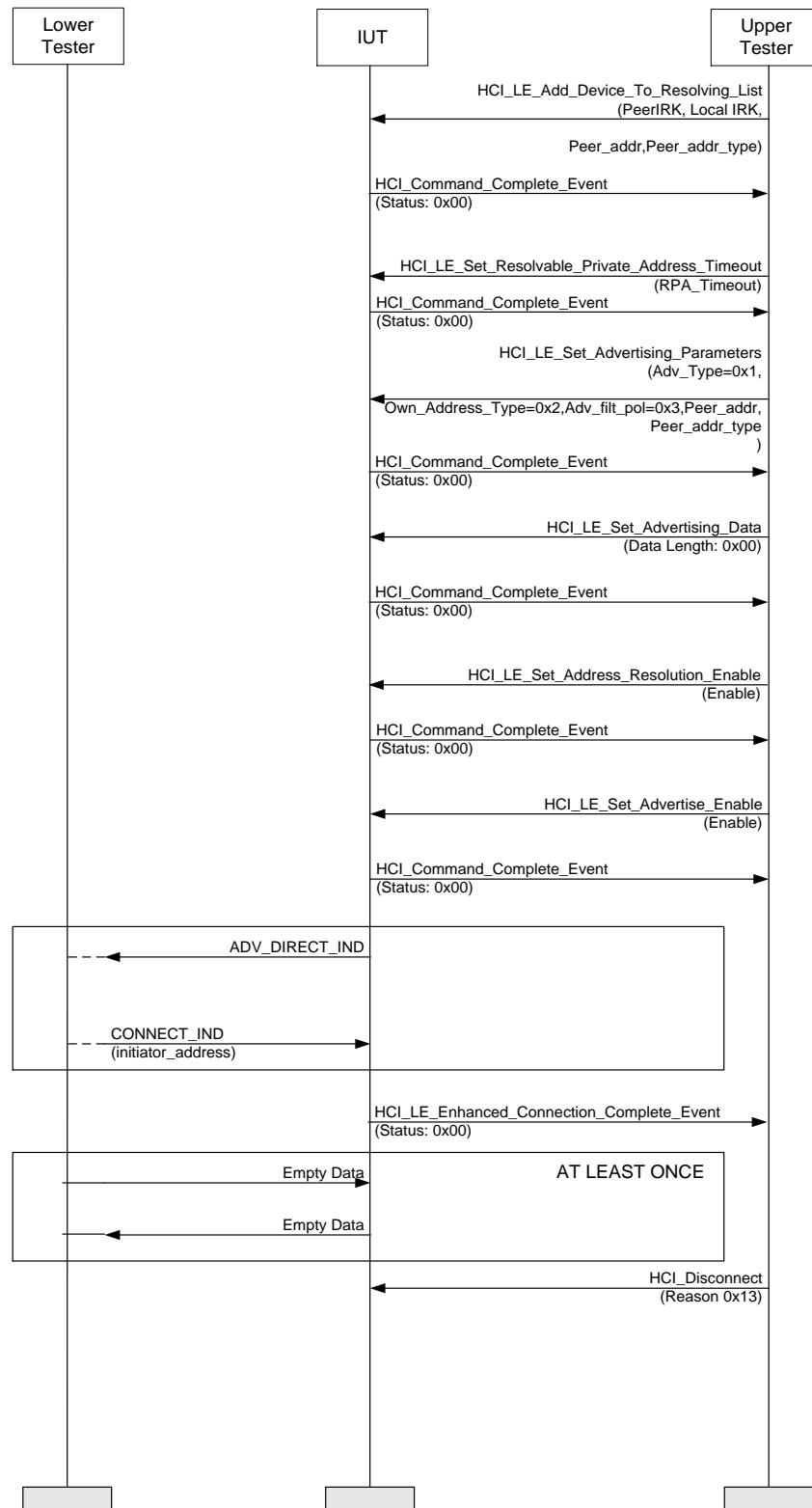


Figure 4.356: LL/SEC/ADV/BV-11-C [Network Privacy - Directed Connectable Advertising using local and remote IRK]

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
 2. Configure Lower Tester to initiate a connection while using a resolvable private address.

3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity. The IUT use these when generating resolvable private addresses for use in the advertising packet's AdvA and InitA fields.
 4. Upper Tester enables resolving list and directed connectable advertising in the IUT.
 5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
 6. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA generated based on its Device Identity. The IUT verifies AdvA and resolves the InitA Address and identifies the Lower Tester.
 7. The Lower Tester connects to the IUT. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The Upper Tester terminates the connection.
- Expected Outcome

Pass Verdict

IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT verifies the AdvA address and resolves the InitA address in the CONNECT_IND packet and accepts the connection.

4.8.2.12 LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK]

- Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, is using resolvable private address for AdvA field and a Public or Static Address (Identity Address) for the InitA field when the Lower Tester has not distributed its own IRK.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

The IUT is using a resolvable private address.

The Lower Tester is using a public or static address and has not distributed its IRK.



- Test Procedure

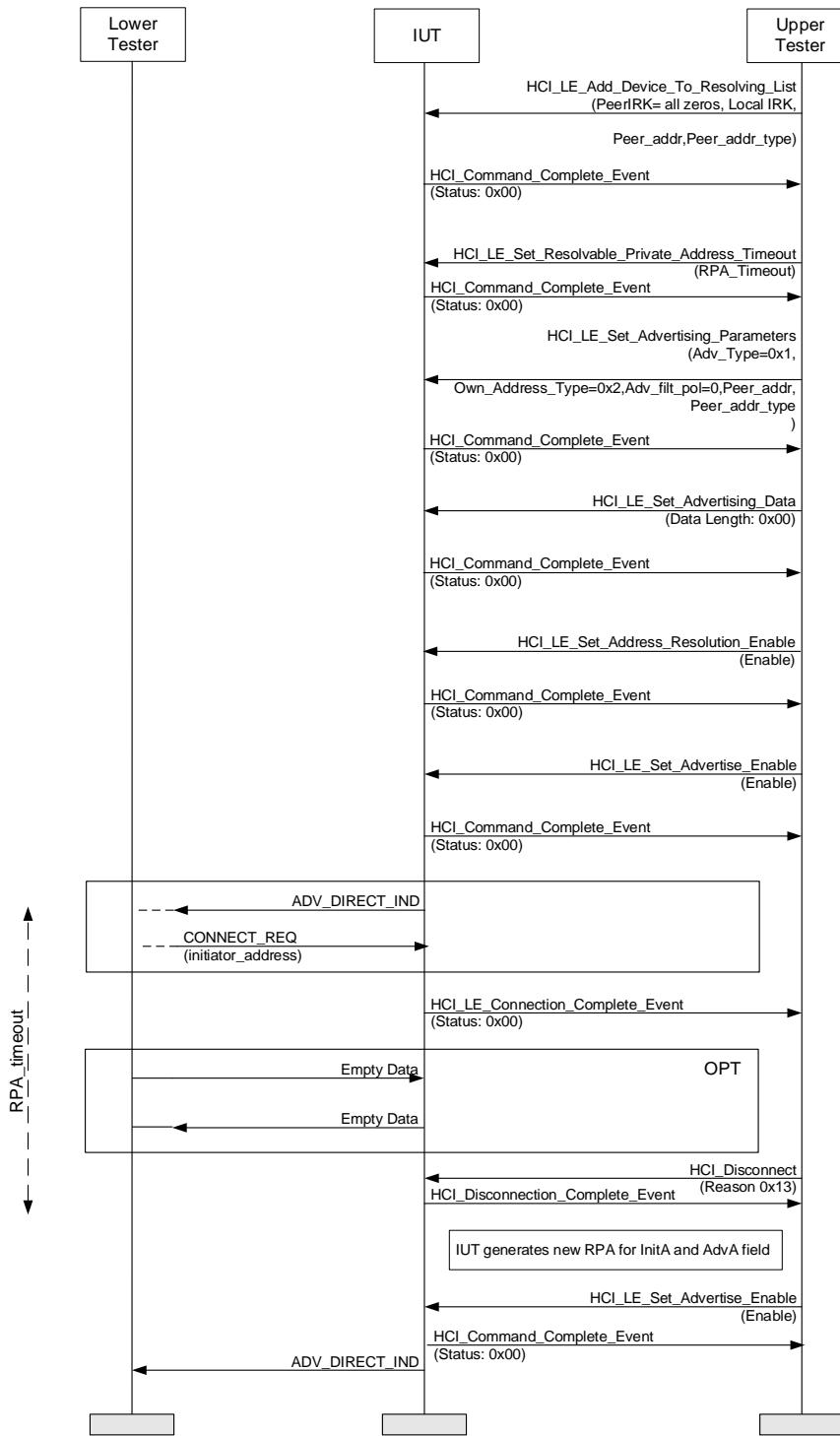


Figure 4.357: LL/SEC/ADV/BV-12-C [Network Privacy - Directed Connectable Advertising with local IRK but without remote IRK]

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure Lower Tester to initiate a connection while using a Public Address.
3. The Upper Tester populates the resolving list with the identity address of the Lower Tester connected with the local device identity. The Lower Tester's IRK is set to all zeros. The IUT uses



- these when generating resolvable private addresses for use in the advertising packet's AdvA field.
4. Upper Tester enables directed connectable advertising in the IUT.
 5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
 6. The Lower Tester resolved the identity of the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA being its Public or Static Random Device Address. The IUT verifies the AdvA address and resolves the InitA address to identify the Lower Tester.
 7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The Upper Tester terminates the connection.
 9. The IUT has generated a new resolvable private address for the AdvA field.
 10. Upper Tester enables connectable directed advertising in the IUT.
 11. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
- Expected Outcome

Pass Verdict

IUT advertises with directed advertising using resolvable private address in the AdvA field.

The IUT accepts connection.

The IUT refreshes the advertiser address after the disconnection.

4.8.2.13 LL/SEC/ADV/BV-13-C [Network Privacy - Directed Connectable Advertising without local IRK but with remote IRK]

- Test Purpose

Verify the IUT when transmitting directed connectable advertising events using a public or static address for AdvA field and a resolvable private address for the InitA field when the Lower Tester has distributed its own IRK.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

The Lower Tester is using a resolvable private address and has distributed its IRK.

The IUT is using a public or static address.



- Test Procedure

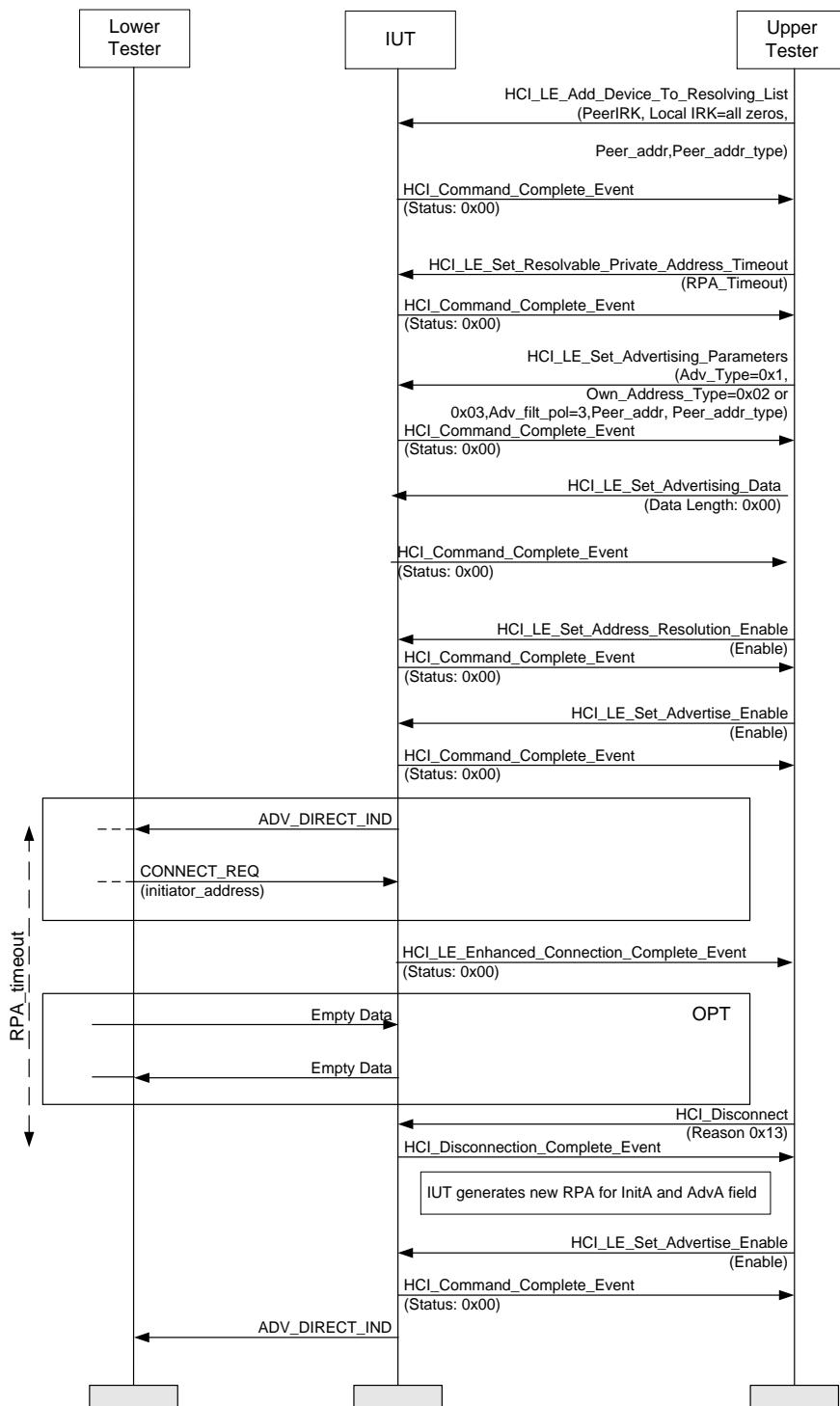


Figure 4.358: LL/SEC/ADV/BV-13-C [Network Privacy - Directed Connectable Advertising without local IRK but with remote IRK]

1. Configure Lower Tester to initiate a connection while using a resolvable private address.
 2. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with a local all zero IRK. The IUT uses this when generating a resolvable private address for use in the advertising packet's InitA field.

3. Upper Tester enables resolving list and directed connectable advertising in the IUT using Own_Address_Type to ResolvableOrPublic (0x02) or ResolvableOrRandom (0x03) according to IUT address (public or static random).
4. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel with initA as resolvable private address.
5. Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA of the ADV_DIRECT_IND and an InitA address generated based on Lower Tester's IRK. The IUT resolves the InitA Address and identifies the Lower Tester.
6. Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
7. The Upper Tester terminates the connection and waits for the RPA Timeout.
8. The IUT has generated a new resolvable private address for the InitA and AdvA field.
9. Upper Tester enables connectable directed advertising in the IUT.
10. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

- Expected Outcome

Pass Verdict

IUT advertises with directed advertising using identity address in the AdvA and RPA in the initA address.

The IUT accepts the connection.

The IUT refreshes the initiator address after the private address refresh timeout.

4.8.2.14 LL/SEC/ADV/BV-14-C [Network Privacy - Directed Connectable Advertising using Resolving List and Peer Device Identity not in the List]

- Test Purpose

Verify the IUT when transmitting directed connectable advertising events and using the Resolving List connects to the devices that are only in the resolving list. The IUT should only connect to a Lower Tester upon successful resolution of the peer's resolvable private address.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

The Lower Tester is using a resolvable private address and has distributed its IRK.

The IUT is using a public or static address.



- Test Procedure

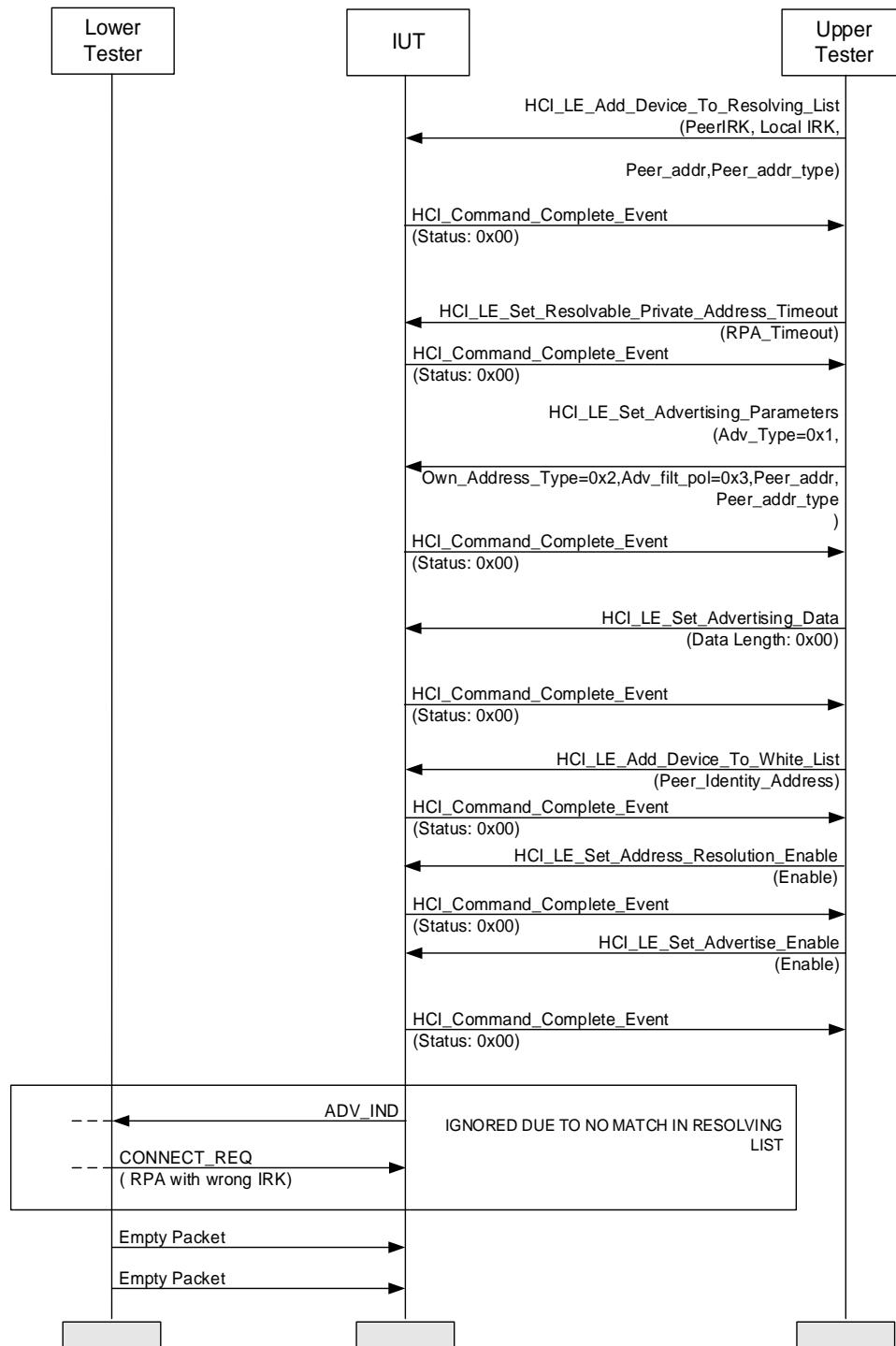


Figure 4.359: LL/SEC/ADV/BV-14-C [Network Privacy - Directed Connectable Advertising using Resolving List and Peer Device Identity not in the List]

- Configure Lower Tester to initiate a connection while using a Resolvable Private Address.
- The Upper Tester populates the resolving list with device identities not equal to the one of the Lower Tester connected with the local device identity.
- The IUT use its device identity when generating a resolvable private address for use in the advertising packet's AdvA field and the device identity in the resolving list for the InitA field.
- Upper Tester enables directed connectable advertising in the IUT.



5. Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
6. Lower Tester sends a CONNECT_IND using the advertiser's address for AdvA field. The InitA field uses a resolvable private address based on the device identity of the Lower Tester (an IRK that is not the same as the one distributed to the Upper Tester).

- Expected Outcome

Pass Verdict

IUT advertises with a resolvable private address in the AdvA field.

The IUT ignores the connect requests from addresses which are not in the resolving list.

IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT resolves the InitA address in the CONNECT_IND packet and does not accept the connection.

4.8.2.15 LL/SEC/ADV/BV-15-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore Identity Address]

- Test Purpose

Verify that an advertiser IUT does not respond to a scan request with the scanner's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

- Reference

[3] 4.4.2.5

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

Execute the test procedure using a single local IRK.

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.



- Test Procedure

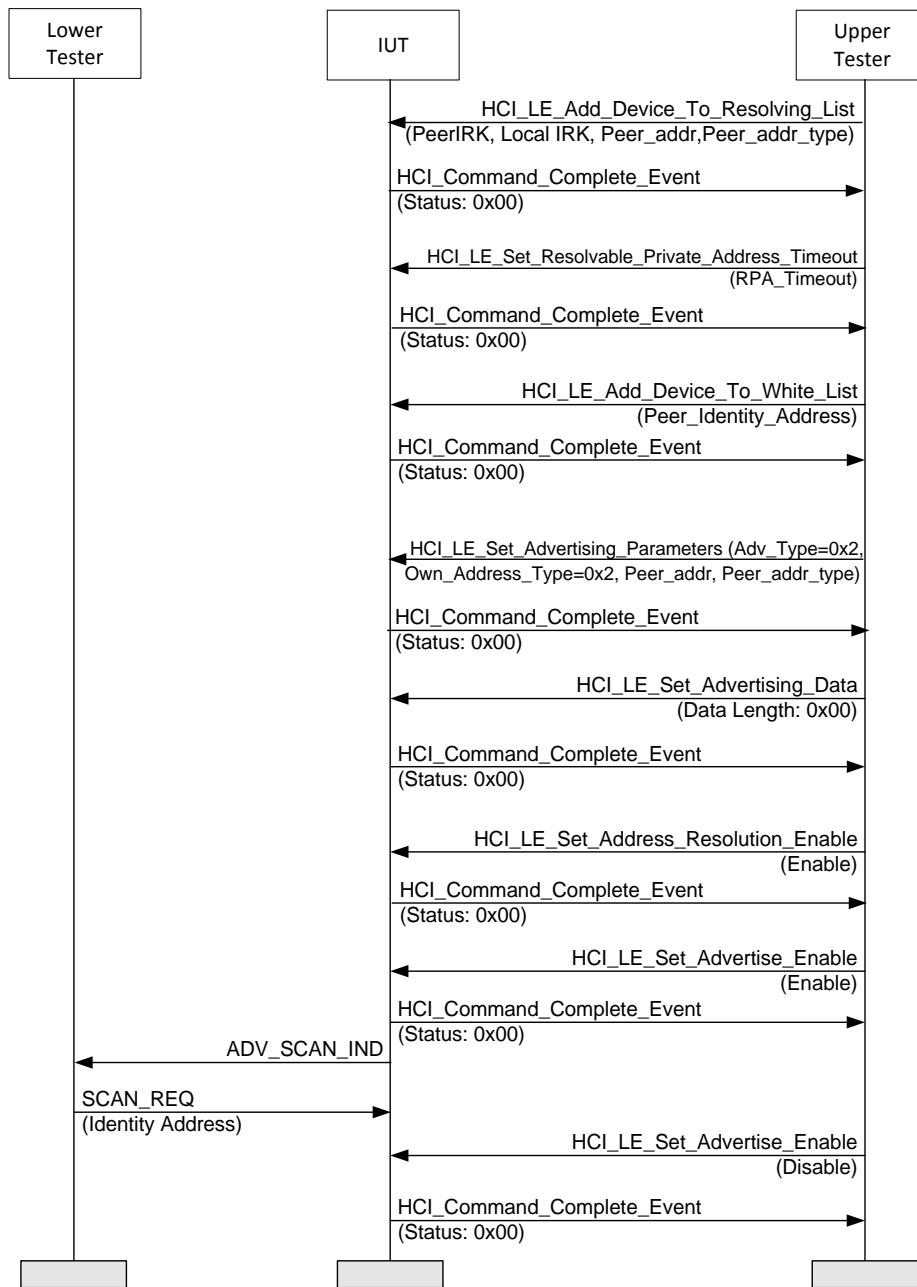


Figure 4.360: LL/SEC/ADV/BV-15-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore Identity Address]

1. Configure the Lower Tester to monitor advertising packets from the IUT.
 2. The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The Advertising_Filter_Policy shall be set to 0x01 (Process connection requests from all devices and only scan requests from devices that are in the White List).
 3. The Upper Tester adds the Lower Tester's Identity Address to the White List.
 4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.

5. The Upper Tester enables scannable undirected advertising.
 6. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
 7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's device identity address.
 8. The IUT compares the address by checking against its resolving list and finds a match with network privacy mode.
 9. The Lower Tester expects the IUT does not send SCAN_RSP packets in response to the SCAN_REQ.
 10. The Upper Tester disables advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

The Lower Tester and IUT successfully resolved the peer's address.

The IUT does not respond to the SCAN_REQ with the Lower Tester's device identity address.

4.8.2.16 LL/SEC/ADV/BV-16-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Ignore Identity Address]

- Test Purpose

Verify that the IUT, when transmitting undirected connectable advertising events, does not connect in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

- Reference

[3] 4.4.2.3

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.



- Test Procedure

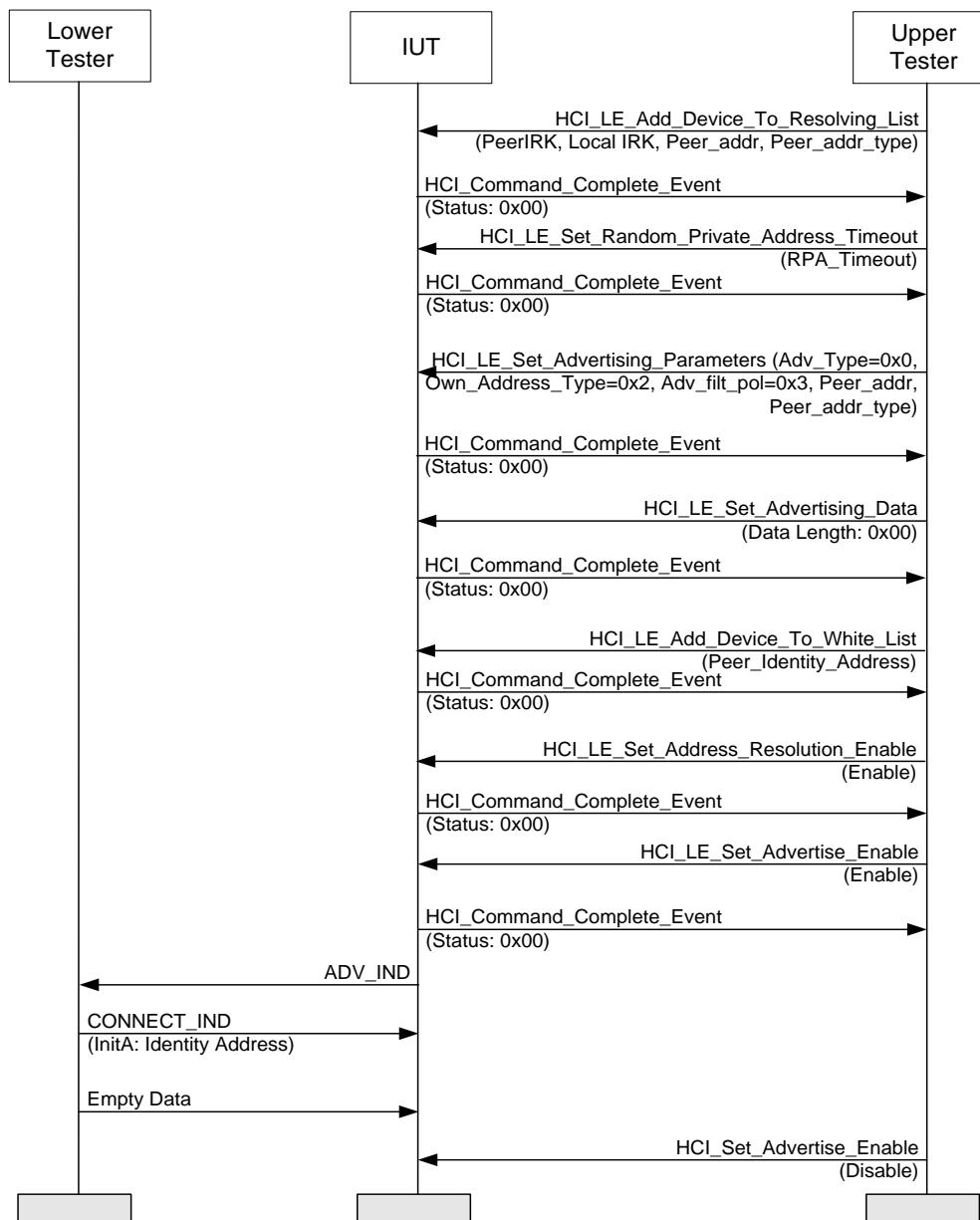


Figure 4.361: LL/SEC/ADV/BV-16-C [Network Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Ignore Identity Address]

- Configure Lower Tester to initiate a connection while using its device identity address.
- The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK. The IUT uses this when generating a resolvable private address for use in the advertisement packet's AdvA field.
- The Upper Tester adds the Lower Tester's Identity Address to the White List.
- The Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
- The Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.



6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA set to its device identity address.
7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The Lower Tester expects the IUT to ignore the CONNECT_IND and expects the IUT not to maintain the connection.
9. The Upper Tester disables advertising.

- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser generating a resolvable private address and using it in connectable advertising.

The IUT ignores the CONNECT_IND and does not maintain the connection.

4.8.2.17 LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address]

- Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, does not connect in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using network privacy mode.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

Lower Tester is using its identity and has previously distributed its IRK.

The IUT is using a resolvable private address.

The IUT is not using the Lower Tester Identity Address in Device Privacy Mode.



- Test Procedure

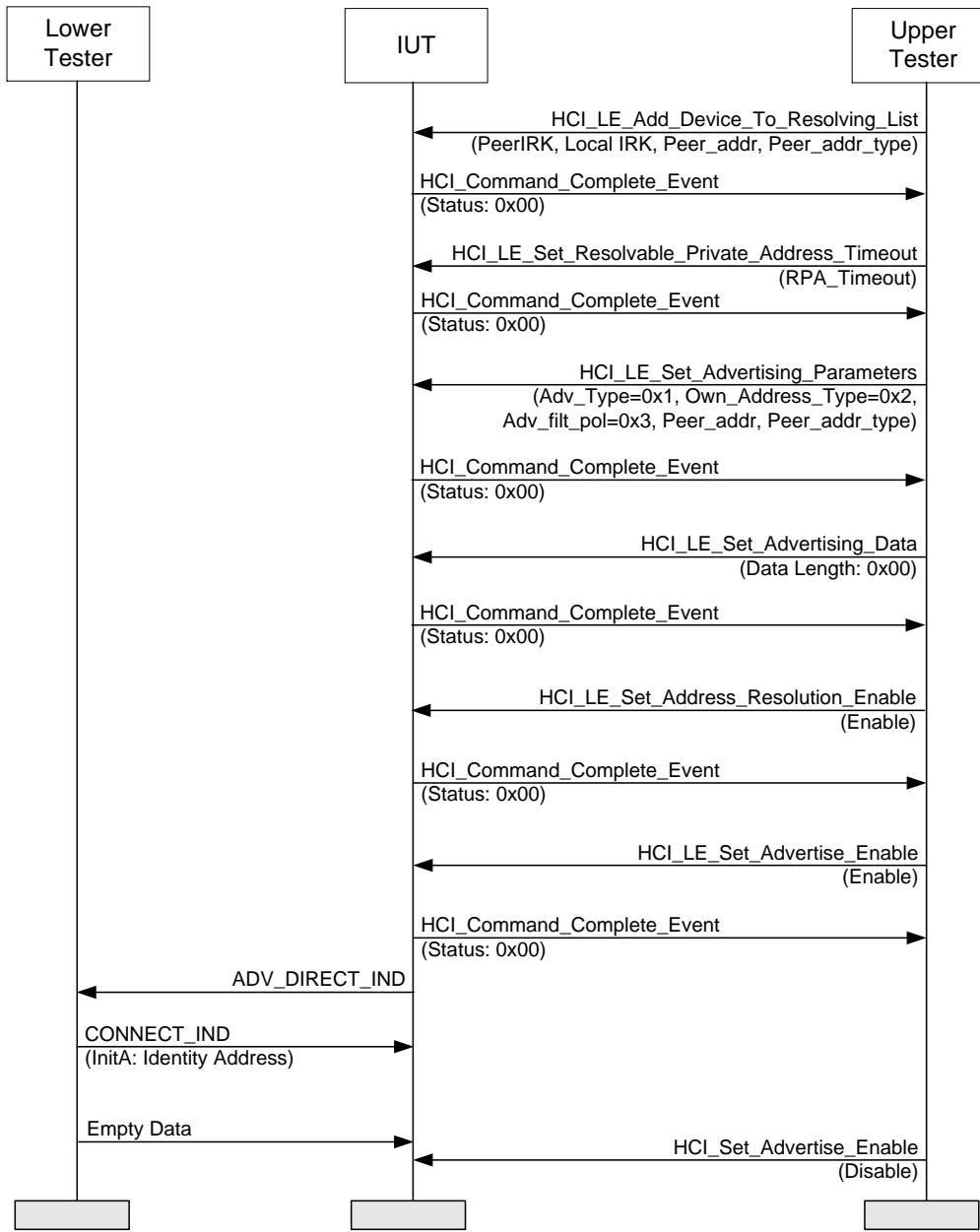


Figure 4.362: LL/SEC/ADV/BV-17-C [Network Privacy - Directed Connectable Advertising using local and remote IRK, Ignore Identity Address]

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure Lower Tester to initiate a connection while using its device identity address.
3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity. The IUT uses this when generating resolvable private addresses for use in the advertisement packet's AdvA and InitA fields.
4. The Upper Tester enables resolving list and directed connectable advertising in the IUT.
5. The Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.

6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_DIRECT_IND and the InitA set to its device identity address.
7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The Lower Tester expects the IUT to ignore the CONNECT_IND and expects the IUT not to maintain the connection.
9. The Upper Tester disables advertising.

- Expected Outcome

Pass Verdict

The IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT ignores the CONNECT_IND and does not maintain the connection.

4.8.2.18 LL/SEC/ADV/BV-18-C [Device Privacy - Scannable Advertising, resolvable private address, Accept Identity Address]

- Test Purpose

Verify that an advertiser IUT responds to a scan request with the scanner's identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

- Reference

[3] 4.4.2.5

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

Execute the test procedure using a single local IRK.



- Test Procedure

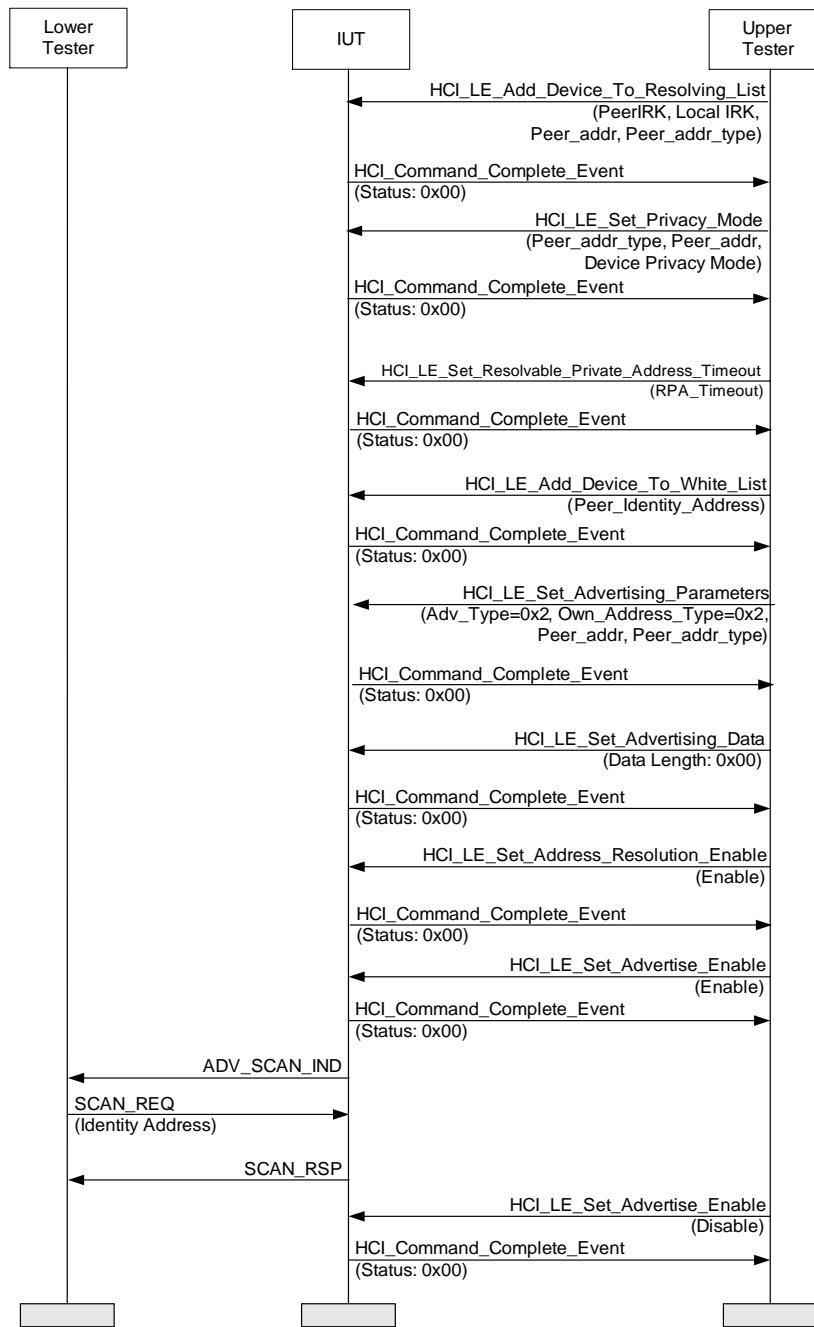


Figure 4.363: LL/SEC/ADV/BV-18-C [Device Privacy - Scannable Advertising, resolvable private address, Accept Identity Address]

1. Configure the Lower Tester to monitor advertising packets from the IUT.
 2. The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 3. The Upper Tester adds the Lower Tester's Identity Address to the White List.
 4. The Upper Tester assigns an IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field. The Lower Tester is assigned the same IRK to be used during address resolution.

5. The Upper Tester enables scannable undirected advertising.
 6. The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
 7. The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's device identity address.
 8. The IUT compares the address by checking against its resolving list and finds a match with device privacy mode.
 9. The Lower Tester expects the IUT to send a SCAN_RSP packet in response to the SCAN_REQ.
 10. The Upper Tester disables advertising.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

The IUT responds to the SCAN_REQ with the advertiser RPA.

4.8.2.19 LL/SEC/ADV/BV-19-C [Device Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Accept Identity Address]

- Test Purpose

Verify that the IUT, when transmitting undirected connectable advertising events, connects in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

- Reference

[3] 4.4.2.3

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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



- Test Procedure

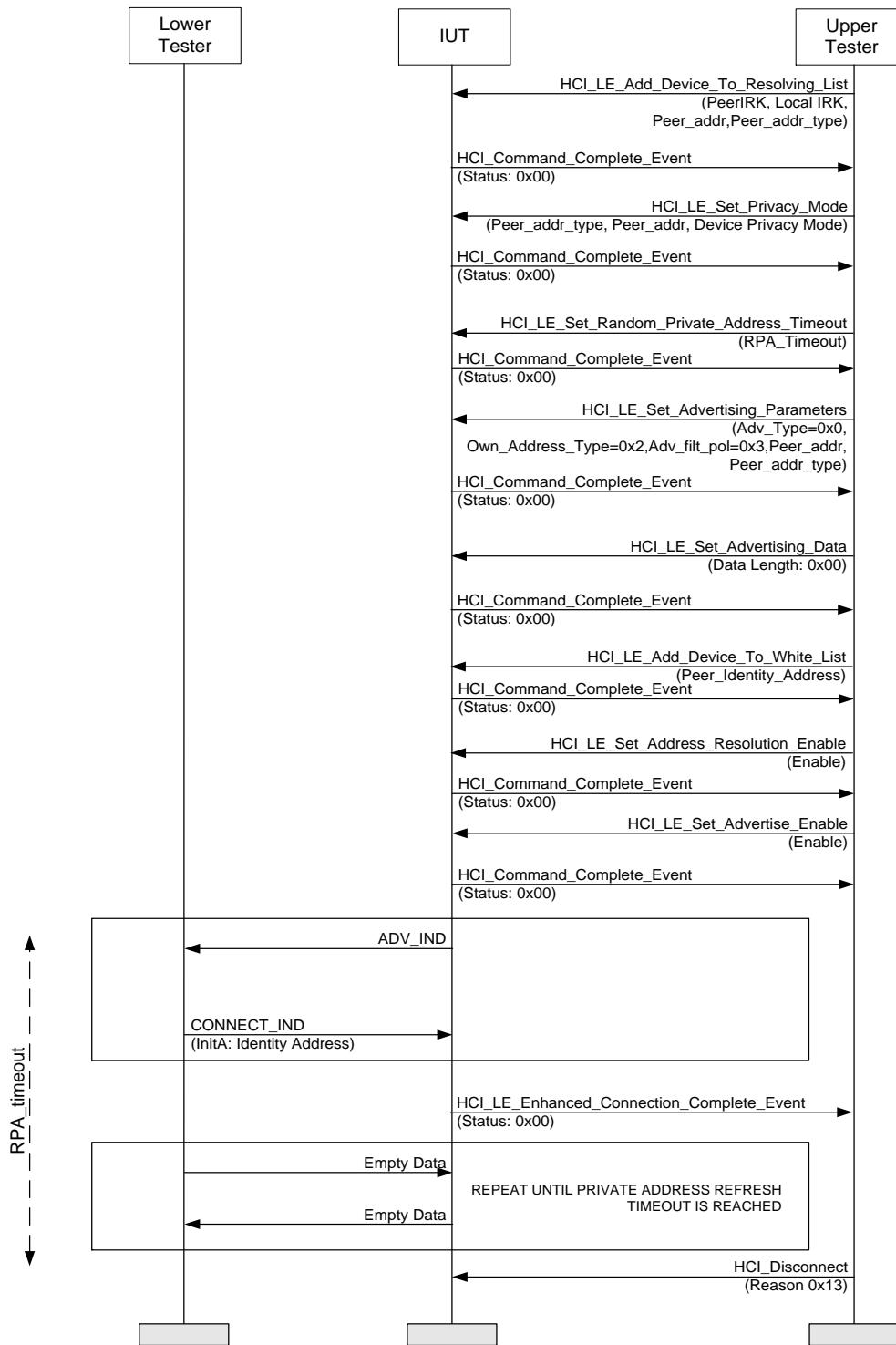


Figure 4.364: LL/SEC/ADV/BV-19-C [Device Privacy - Undirected Connectable Advertising with Local IRK and Peer IRK, Accept Identity Address]

1. Configure the Lower Tester to initiate a connection while using its device identity address.
 2. The Upper Tester populates the resolving list with the device identity (IRK and Identity Address) of the Lower Tester and the local IRK and sets the entry to device privacy mode. The IUT uses

- this when generating a resolvable private address for use in the advertisement packet's AdvA field.
3. The Upper Tester adds the Lower Tester's Identity Address to the White List.
 4. The Upper Tester enables connectable advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals.
 5. The Lower Tester expects the IUT to send ADV_IND packets on an applicable advertising channel.
 6. The Lower Tester resolves the AdvA address and identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_IND and the InitA set to its device identity address. The IUT verifies the AdvA field and compares the InitA address by checking against its resolving list and finds a match with device privacy mode.
 7. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
 8. The connection is maintained for a time interval greater than or equal to the timeout value for the private address refresh.
 9. The IUT terminates the connection.
- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser generating a resolvable private address, using it in connectable advertising, and connecting to the Lower Tester.

4.8.2.20 LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address]

- Test Purpose

Verify that the IUT, when transmitting directed connectable advertising events, connects in response to connect requests with the initiator's identity address when the IUT has that address and an associated IRK in the resolving list using device privacy mode.

- Reference

[3] 4.4.2.4

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_Adv_Channel_Map

State: Directed Advertising (supported type of own address, public initiator address, Lower Tester address, selected advertising channels)

Lower Tester is using its identity address and has previously distributed its IRK.

The IUT is using a resolvable private address.



- Test Procedure

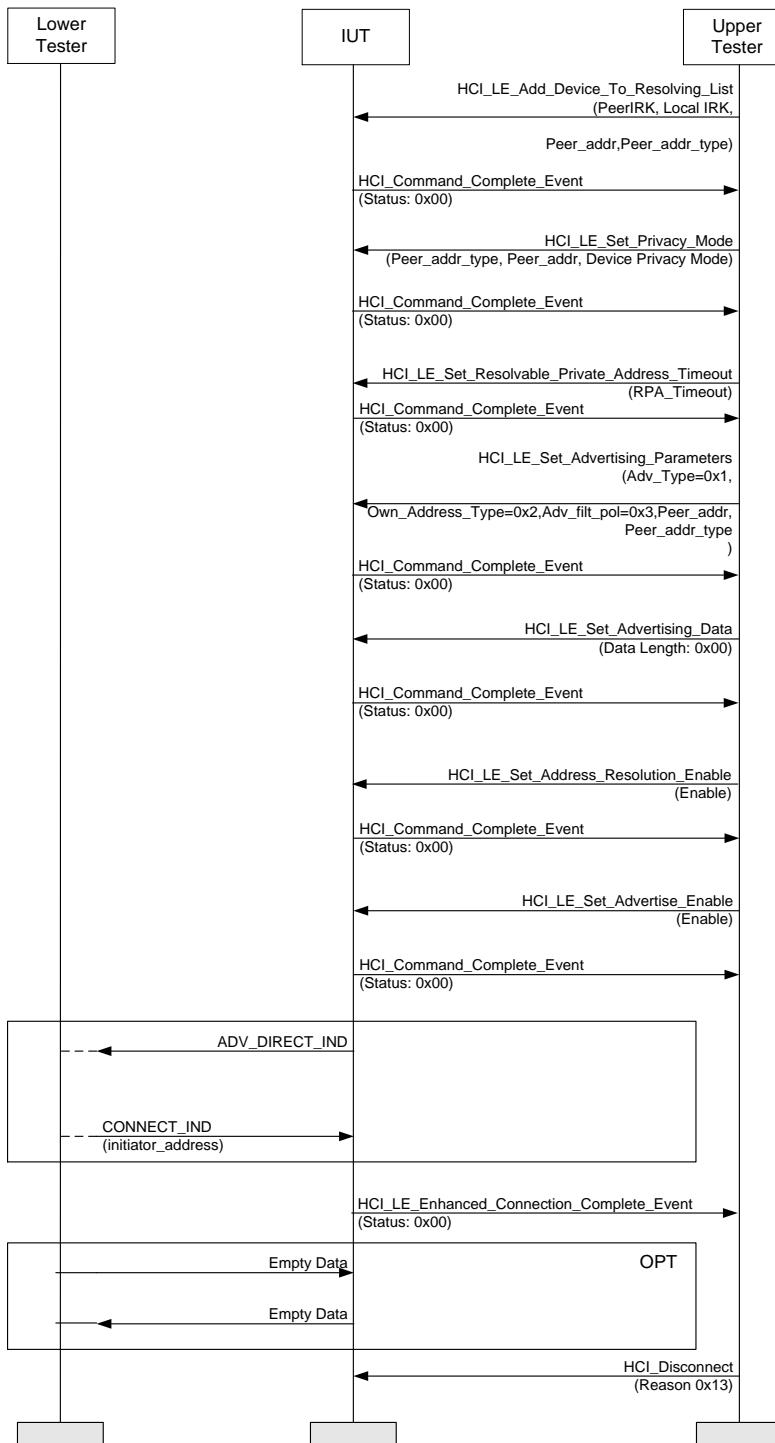


Figure 4.365: LL/SEC/ADV/BV-20-C [Device Privacy - Directed Connectable Advertising using local and remote IRK, Accept Identity Address]

1. The Lower Tester adds the Device Identity of the IUT to its resolving list.
2. Configure the Lower Tester to initiate a connection while using its device identity address.
3. The Upper Tester populates the resolving list with the device identity of the Lower Tester connected with the local device identity and sets the entry to device privacy mode. The IUT uses



this when generating resolvable private addresses for use in the advertisement packet's AdvA and InitA fields.

4. The Upper Tester enables resolving list and directed connectable advertising in the IUT.
5. The Lower Tester expects the IUT to send ADV_DIRECT_IND packets on an applicable advertising channel.
6. The Lower Tester identifies the IUT. The Lower Tester sends a CONNECT_IND with the AdvA address of the ADV_DIRECT_IND and the InitA set to its device identity address.
7. The Lower Tester connects to the IUT. The Lower Tester sends empty LL DATA packets starting with the first event one connection interval after the connection request using the common data channel selection parameters.
8. The IUT terminates the connection.

- Expected Outcome

Pass Verdict

IUT advertises with directed advertising using resolvable private addresses in the AdvA and InitA fields.

The IUT verifies the AdvA address and compares the InitA address in the CONNECT_IND packet by checking against its resolving list and finds a match with device privacy mode and accepts the connection

4.8.2.21 LL/SEC/ADV/BV-21-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA]

- Test Purpose

Verify that an advertiser IUT does not respond to a scan request with the scanner's RPA when the IUT has that address and an associated IRK in the resolving list using network privacy mode but address resolution is not enabled.

- Reference

[3] 4.4.2.5

- Initial Condition

The Upper Tester configures the Resolvable Private Address Timeout to the value defined in the IXIT.

Parameters: LL_advertiser_advInterval_MIN, LL_advertiser_advInterval_MAX,
LL_advertiser_Adv_Channel_Map

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

Execute the test procedure using a single local IRK.



- Test Procedure

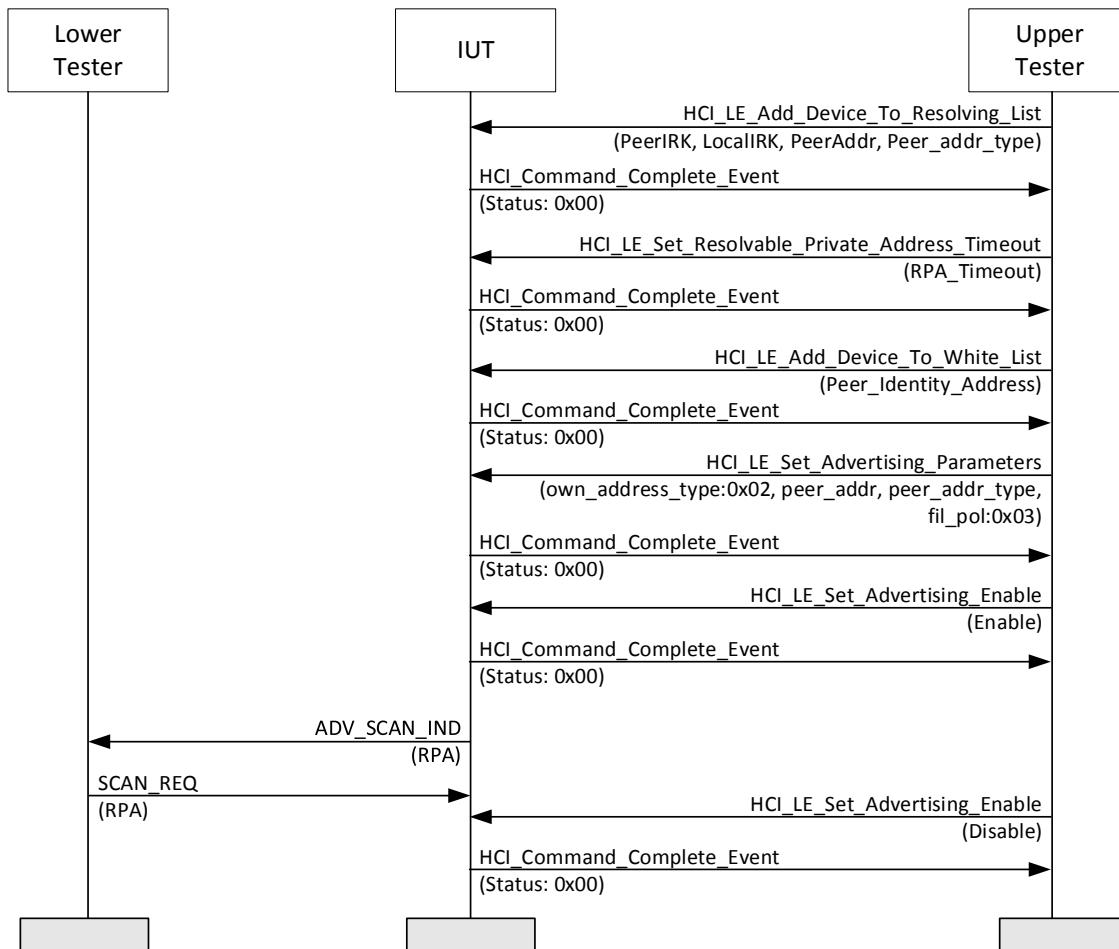


Figure 4.366: LL/SEC/ADV/BV-21-C [Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA]

- Configure the Lower Tester to monitor advertising packets from the IUT.
- The Upper Tester assigns a valid Local IRK to the IUT to be used in generating a resolvable private address for use in the advertisement packets AdvA field using add device to resolving list command.
- The Upper Tester adds the Lower Tester's Identity Address to the White List.
- The Upper Tester enables scannable undirected advertising in the IUT using all supported advertising channels and a selected advertising interval between the minimum and maximum advertising intervals. The Advertising_Filter_Policy shall be set to 0x03 (Process scan requests and connect requests from devices that are in the White List).
- The Lower Tester expects the IUT to send ADV_SCAN_IND packets on an applicable advertising channel. Lower Tester resolves the private address received from the IUT using assigned IRK.
- The Lower Tester sends a SCAN_REQ packet for every received ADV_SCAN_IND packet from the IUT with ScanA set to the Lower Tester's RPA.
- The IUT compares the lower tester's address and does not try to resolve the lower tester's RPA.
- The Lower Tester expects the IUT does not send SCAN_RSP packets in response to the SCAN_REQ.
- The Upper Tester disables advertising.

- Expected Outcome

Pass Verdict

The test procedure is executed successfully, with the advertiser using a resolvable private address generated from the assigned IRK, using it in scannable advertising.

The IUT does not respond to the SCAN_REQ with the Lower Tester's RPA.

4.8.3 SCN

Tests that the IUT handles random addresses in the scanner role.

4.8.3.1 LL/SEC/SCN/BV-01-C [Random Address Scanning]

- Test Purpose

Test that a scanner IUT is able to calculate and use random addresses. Test that the IUT responds with Command Disallowed to an LE Set Random Address command when scanning is enabled.

- Reference

[3] 5.1.3, 4.4.3.2

- Initial Condition

State: Random Address Calculated (common ir) AND Active Scanning (random address, selected scan interval, selected scan window) AND All White Listed (policy for scanner)

- Test Procedure

Execute the test procedure using a single random address as the advertiser address. The test uses the common variable 'ir' to encrypt and resolve random addresses. The preambles steps calculate the identity resolving key (variable 'irk').



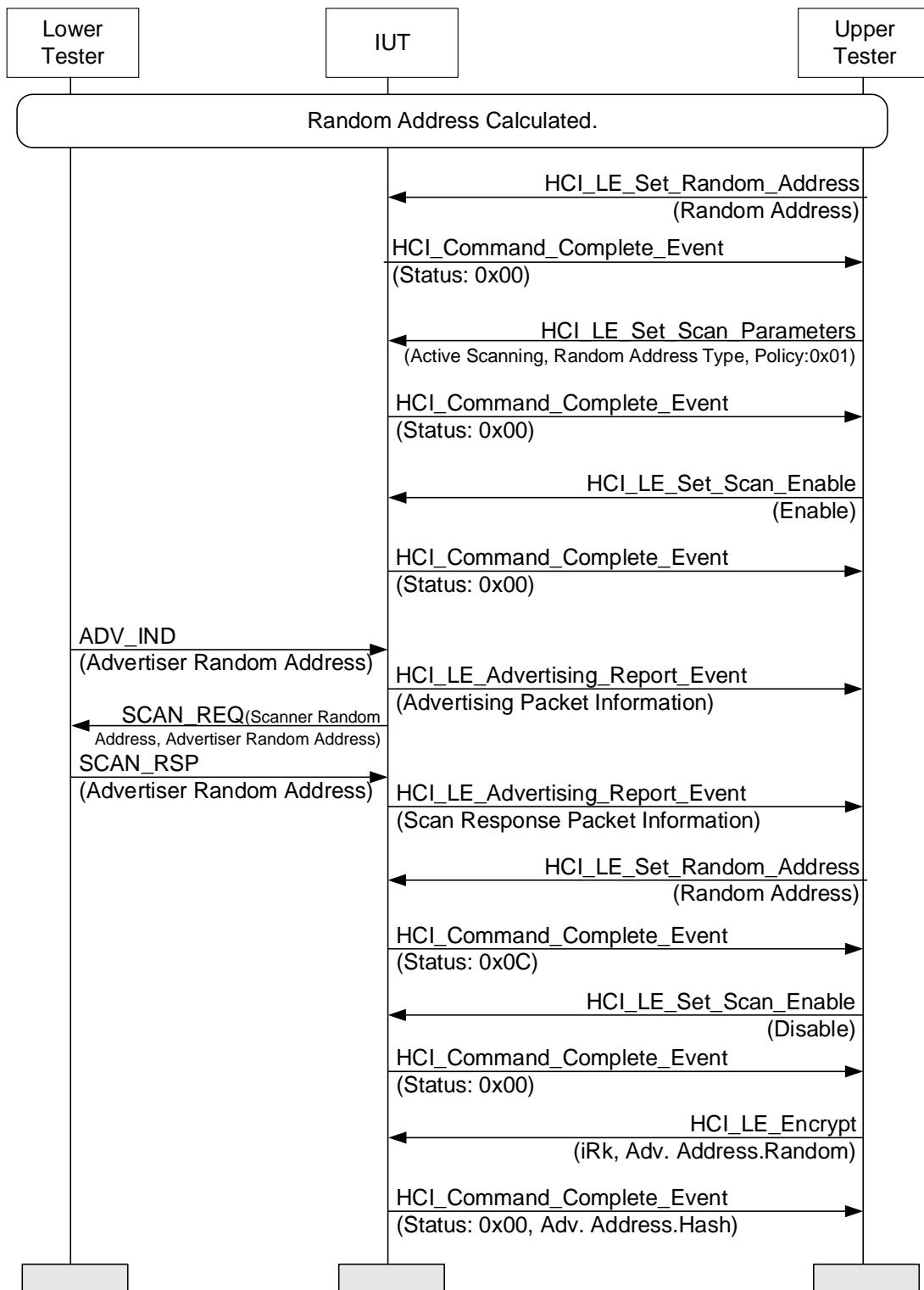


Figure 4.367: LL/SEC/SCN/BV-01-C [Random Address Scanning]

1. Upper Tester sends an **HCI_LE_Set_Random_Address** command to set the IUT random address.
2. Upper Tester enables active scanning in the IUT.
3. Configure the Lower Tester the advertising with random device address.

4. Lower Tester sends an ADV_IND packet with the private address, each advertising event on the selected advertising channel only, using the selected advertising interval.
 5. Lower Tester expects an SCAN_REQ with the Lower Tester random address and the IUT random address in response after TIFS to any of the packets.
 6. Lower Tester sends an SCAN_RSP containing the random address to the IUT TIFS after of the response.
 7. Upper Tester expects an HCI_LE_Advertising_Report event containing the advertising packet information.
 8. Upper Tester expects an HCI_LE_Advertising_Report event from the IUT containing the address in the response sent in step 6.
 9. Upper Tester sends an HCI_LE_Set_Random_Address command to set the IUT random address and expects an HCI_Command_Complete event from the IUT with a Status of 0x0C.
 10. Upper Tester sends an HCI_LE_Set_Scan_Enable to the IUT to disable scanning and expects an HCI_Command_Complete event in response.
 11. Upper Tester sends an HCI_LE_Encrypt command to the IUT with the parameters 'irk' and 3 least significant octets of the advertiser address received. Expect an HCI_Command_Complete event in response with the result parameter matching to the 3 most significant octets of the advertiser address received.
- Expected Outcome

Pass Verdict

The IUT generates a random address.

The IUT uses the random address in the SCAN_REQ packets.

The IUT resolves a random address received in the ADV_IND packets.

The IUT rejects the second HCI_LE_Set_Random_Address command.

4.8.4 INI

Tests that the IUT behaves according to the security procedures in the initiator role.

4.8.5 SLA

Tests that the IUT behaves according to the security procedures in the slave role.

4.8.5.1 LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup]

- Test Purpose

Test that a slave IUT can perform the encryption mode change procedure, using the correct encryption.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption mode setup.
- Reference

[3] 5.1.3
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)



State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (common variables 'er' and 'ir').

The test procedure uses the common variables 'ir' and 'er' to calculate a diversifier hiding key ('dhk'), to derive an encrypted diversifier ('ediv'), to check the diversifier and finally to derive a long term key ('ltk') to set to the slave device.



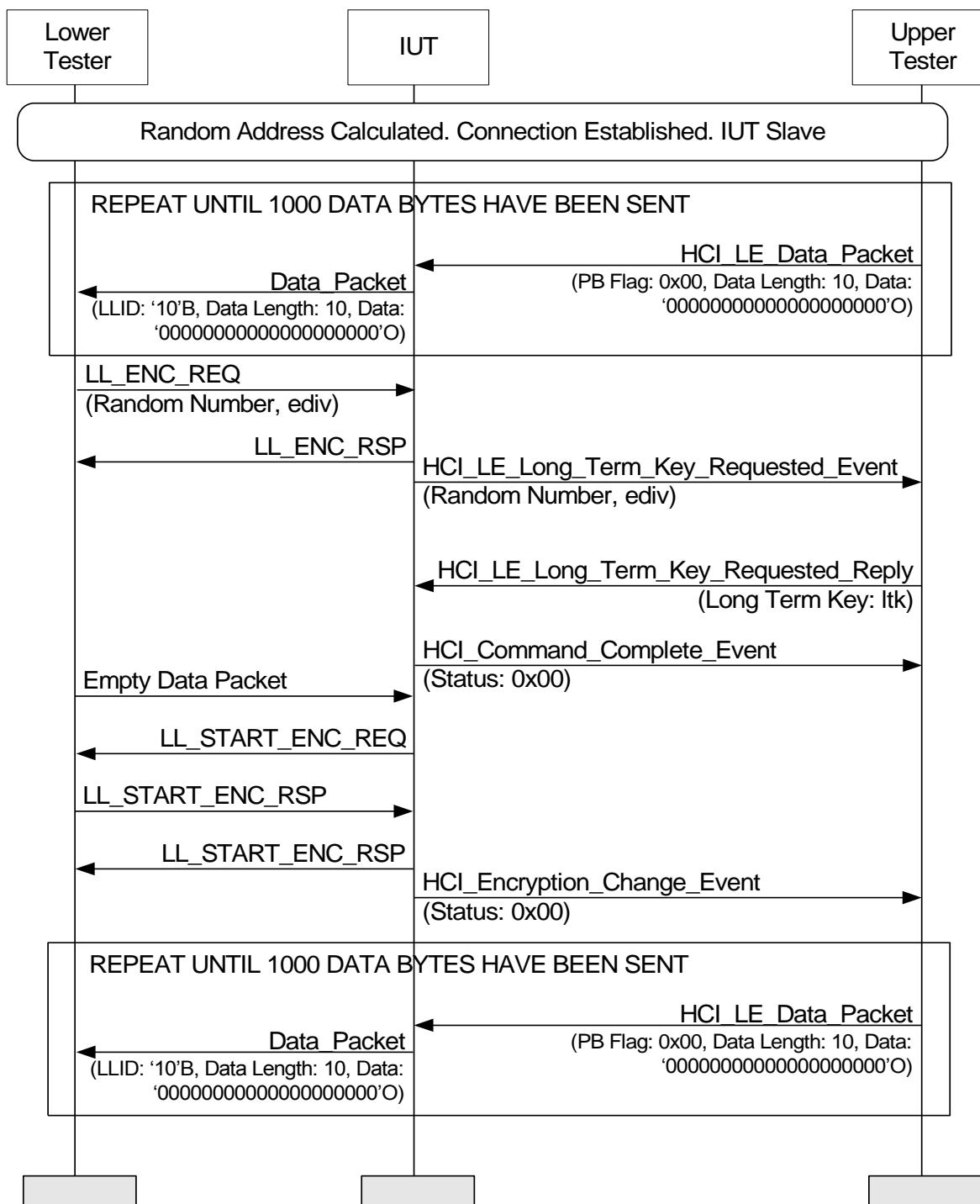


Figure 4.368: LL/SEC/SLA/BV-01-C [Slave Encryption Mode Setup]

1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
2. Configure Upper Tester to submit data elements to the IUT with the HCI LE Data Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
3. Lower Tester sends empty DATA packet once a connection interval to the IUT and expects a DATA packet in response from the IUT until all data sent in 2 have been received.

4. In the Lower Tester, calculate the master portions of the session key diversifier, the initialization vector and a random number.
 5. Lower Tester sends an ENC_REQ packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector to the IUT, expect a LL_ENC_RSP in response. Continue the master transmissions.
 6. Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 5.
 7. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply with 'ltk' to the IUT and expects an HCI_Command_Complete in response.
 8. Lower Tester expects an START_ENC_REQ packet in response to the master transmissions and acknowledges this packet.
 9. Lower Tester sends a START_ENC_RSP packet encrypted to the IUT and expects a LL_START_ENC_RSP packet encrypted in response.
 10. Upper Tester expects an HCI_Encryption_Enable event with encryption enable set to on
 11. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
 12. Lower Tester sends empty DATA packet once a connection interval to the IUT and expects the DATA packets from the IUT to contain encrypted payloads with matching MIC fields until all data sent in step 11 has been received.
 13. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle).
 14. In the preamble steps (connection establishment use random address type for the Lower Tester and a random type of address for the IUT.
 15. Repeat steps 2-12.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the data correctly transmitted and reported by the IUT.

The IUT sends its initialization vector and session key diversifier in a LL_ENC_RSP packet.

The IUT reports the encryption mode setup requested with the HCI event.

The IUT sends a LL_START_ENC_REQ packet until acknowledged.

The IUT acknowledges the LL_START_ENC_RSP and responds with one.

The IUT reports the encryption mode setup with the HCI event.

4.8.5.2 LL/SEC/SLA/BV-02-C [Slave Pause Encryption]

- Test Purpose

Test that a slave IUT can perform the encryption pause procedure.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption pause procedure.

- Reference

[3] 5.1.3.2



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

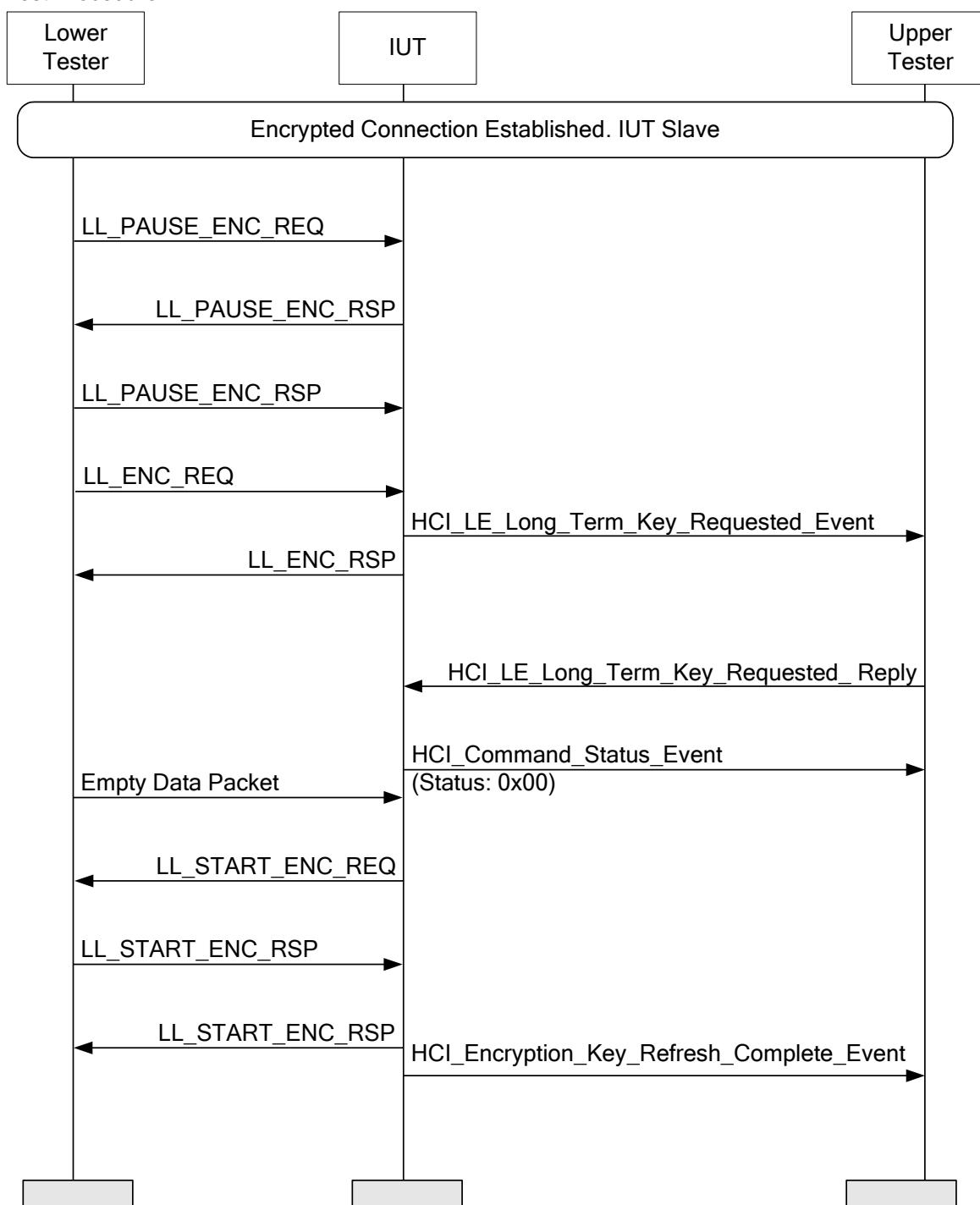


Figure 4.369: LL/SEC/SLA/BV-02-C [Slave Pause Encryption]



1. As preamble: an encrypted connection is established (IUT acts as slave)
 2. Lower Tester sends an encrypted LL_PAUSE_ENC_REQ packet and expects an encrypted LL_PAUSE_ENC_RSP packet in response.
 3. When the Lower Tester receives the LL_PAUSE_ENC_RSP packet, it responds with an unencrypted LL_PAUSE_ENC_RSP. The connection is not encrypted at this point.
 4. Lower Tester sends an LL_ENC_REQ packet to re-enable encryption using a new session key and expect an LL_ENC_RSP packet in response.
 5. Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 4.
 6. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply with 'ltk' to the IUT, expect an HCI_Command_Complete in response.
 7. Lower Tester expects an START_ENC_REQ in response to the master transmissions and acknowledges the packet.
 8. Lower Tester sends an START_ENC_RSP packet encrypted to the IUT and expects a LL_START_ENC_RSP packet encrypted in response.
 9. Upper Tester expects an HCI_Encryption_Key_Refresh_Complete_Event event with encryption enable set to on.
 10. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle).
- Expected Outcome

Pass Verdict

The IUT pauses encryption.

The IUT sends LL_START_ENC_REQ packet.

The IUT sends HCI_Encryption_Key_Refresh_Complete_Event once encryption is resumed.

4.8.5.3 LL/SEC/SLA/BV-03-C [Slave Pause Encryption Sending Data]

- Test Purpose

Test that a slave IUT does not send any data packet during the encryption pause procedure.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption pause procedure and submits data for the slave to transmit. The IUT should not send the data packet until encryption is resumed.

- Reference

[3] 5.1.3.2

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value) State for executing with encrypted private addresses: Encrypted Address Calculated (common ir) AND Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).



- Test Procedure



Figure 4.370: LL/SEC/SLA/BV-03-C [Slave Pause Encryption Sending Data]

- As preamble: an encrypted connection is established (IUT acts as slave).
- Lower Tester sends an encrypted LL_PAUSE_ENC_REQ packet and expects an encrypted LL_PAUSE_ENC_RSP packet in response.



3. When Lower Tester receives the LL_PAUSE_ENC_RSP packet, responds unencrypted LL_PAUSE_ENC_RSP and now the connection is not encrypted.
 4. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and one data elements with the value 0xFF.
 5. Lower Tester sends an LL_ENC_REQ packet to re-enable encryption using a new session key and expects an LL_ENC_RSP packet in response.
 6. Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 4.
 7. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Reply command with 'ltk' to the IUT, expect an HCI_Command_Complete in response.
 8. Lower Tester expects an START_ENC_REQ packet in response to the master transmissions. Acknowledge the packet.
 9. Lower Tester sends an START_ENC_RSP packet encrypted to the IUT and expects a LL_START_ENC_RSP packet encrypted in response.
 10. Upper Tester expects an HCI_Encryption_Key_Refresh_Complete event with encryption enable set to on
 11. Lower Tester sends an empty DATA packet once a connection interval to the IUT and expects the DATA packet from the IUT to contain an encrypted payload with matching MIC field
 12. Slave Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
- Expected Outcome

Pass Verdict

The IUT sends the data packet once encryption is resumed.

4.8.5.4 LL/SEC/SLA/BV-04-C [Slave Sending Reject_Ind]

- Test Purpose

Test that, while executing the Encryption Start Procedure, a slave IUT sends a correct Reject_Ind control packet if the Host does not provide a Long Term Key.

The Lower Tester acts as master, maintaining a connection, then initiates the encryption procedure, failing to provide a Long Term Key to the slave IUT.

- Reference

[3] 5.1.3.1

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).

Lower Tester does not support LL_REJECT_EXT_IND.

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and "ir").



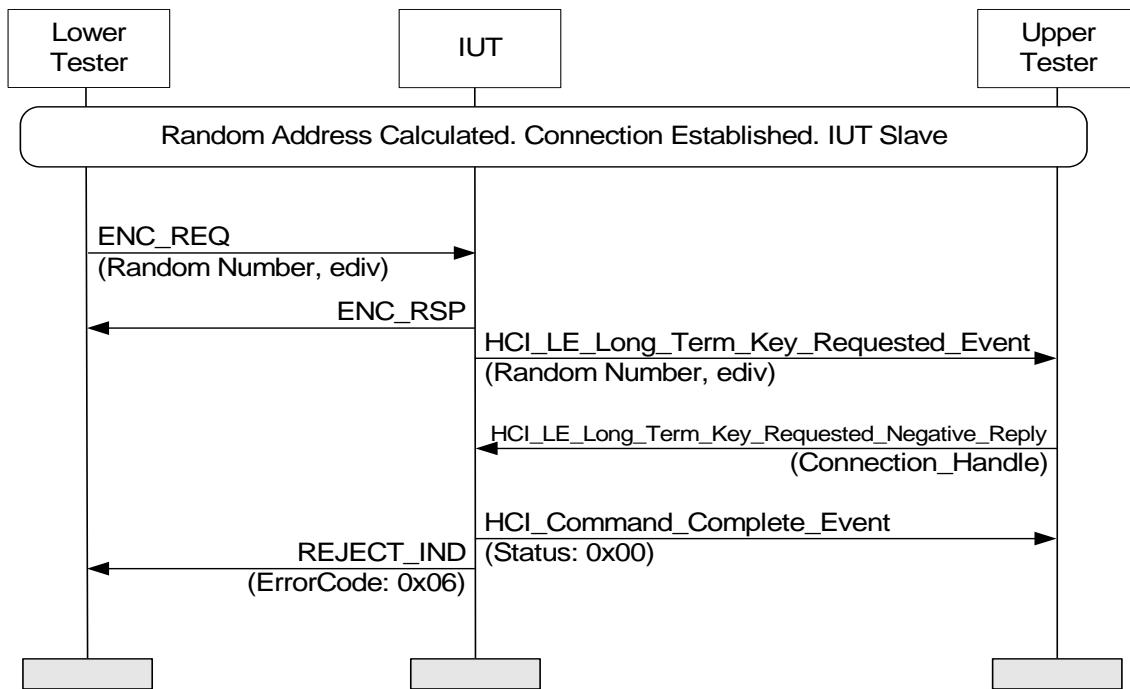


Figure 4.371: LL/SEC/SLA/BV-04-C [Slave Sending Reject_Ind]

1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
 2. Upper Tester sends an HCI_LE_Encrypt command to the IUT, with the key 'ir' and number 2 and expects an HCI_Command_Complete containing the 'dhk' in response.
 3. Lower Tester sends an ENC_REQ packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector to the IUT. Expect an ENC_RSP packet in response and continue the master transmissions.
 4. Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 3.
 5. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Negative_Reply command to the IUT and expects an HCI_Command_Complete in response.
 6. Lower Tester expects a REJECT_IND packet from the IUT with ErrorCode indicating "PIN or Key missing".
- Expected Outcome

Pass Verdict

Lower Tester receives an ENC_RSP packet in response to the ENC_REQ packet it had previously sent to the IUT.

Upper Tester receives an HCI_Command_Complete event with the Status parameter set to 'Success' in response to the HCI_LE_Long_Term_Key_Requested_Negative_Reply command.

Lower Tester receives a REJECT_IND packet from the IUT with ErrorCode indicating 'PIN or key missing'.

4.8.5.5 LL/SEC/SLA/BV-05-C [Slave Receiving Encrypted Data]

- Test Purpose

Test that a slave IUT can receive encrypted data.



The Lower Tester acts as a master, maintaining an encrypted connection, then initiates sending encrypted data packets without fragmentation.

- Reference

[3] 5.1.3

- Initial Condition

State: Connected Slave (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

State for executing with random addresses: Random Address Calculated (common ir) AND Connected Slave (any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

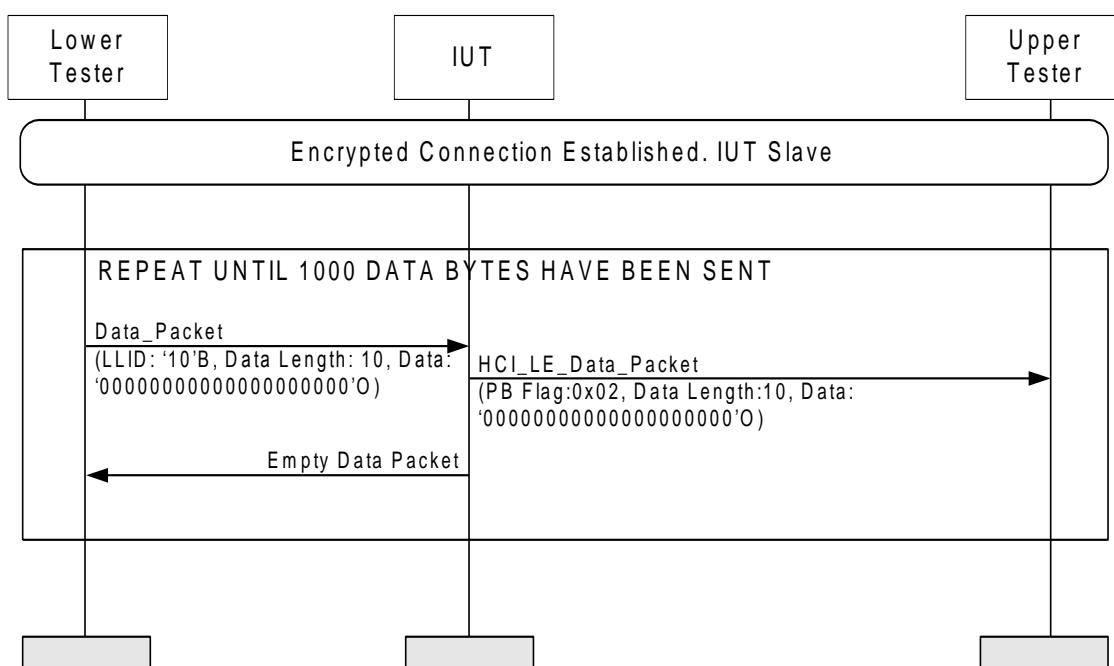


Figure 4.372: LL/SEC/SLA/BV-05-C [Slave Receiving Encrypted Data]

1. Configure Lower Tester to send 1000 encrypted data packets with the LLID field set to 0x02 and using a payload length of 10 with the payload octets set to 0x00.
2. Lower Tester sends a DATA packet once a connection interval to the IUT, using the acknowledgement scheme and the data channel selection parameters, with the LLID field set to 0x02, using a payload length of 10 with the payload octets set to 0x00. Lower Tester expects an empty DATA packet in response from the IUT.
3. Upper Tester expects an HCI_LE_Data_Packet event from the IUT containing a data element sent in step 1 with the Packet_Boundary_Flag flag set to 0x02.
4. Repeat steps 2-3 until all data sent in step 1 has been reported.

- Expected Outcome

Pass Verdict

The test procedure completes with the IUT acknowledging all the data sent,



The IUT reports all data correctly with HCI_LE_Data_Packet events using the HCI fragmentation flags, as specified in Section 4.1.6.

4.8.5.6 LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled]

- Test Purpose

Verify that the IUT as slave sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_PING_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

- Reference

[3] 5.1.8

- Initial Conditions

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.



- Test Procedure

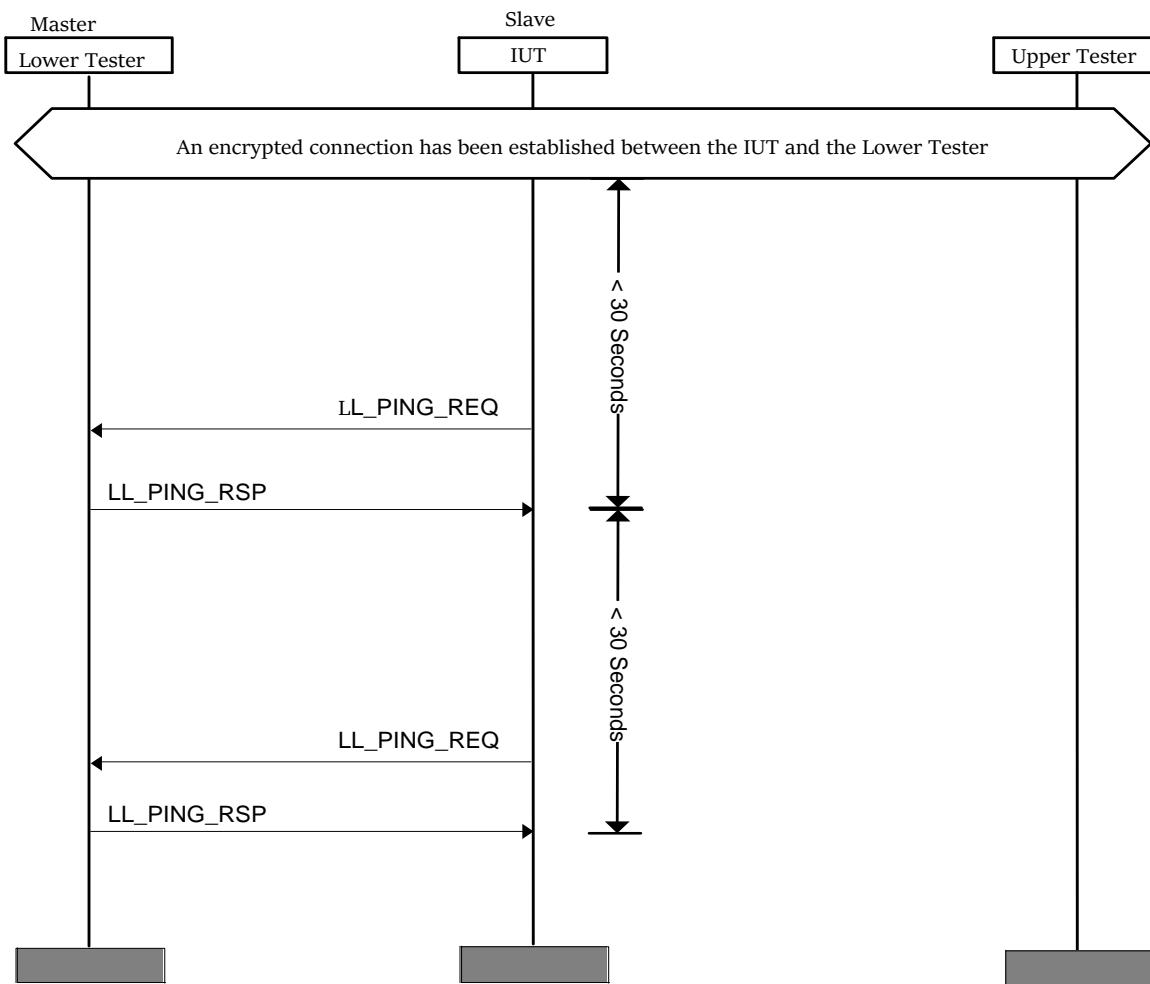


Figure 4.373: LL/SEC/SLA/BV-06-C [Initiate LE Ping procedure when encryption is enabled]

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_PING_RSP.
2. The Lower Tester responds with LL_PING_RSP.

- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.

- Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.5.7 LL/SEC/SLA/BV-07-C [Responding to LL_PING_REQ]

- Test Purpose

Verify that the IUT as slave responds to an LL_PING_REQ sent by the Lower Tester.



- Reference
[3] 5.1.8
- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged

- Test Procedure

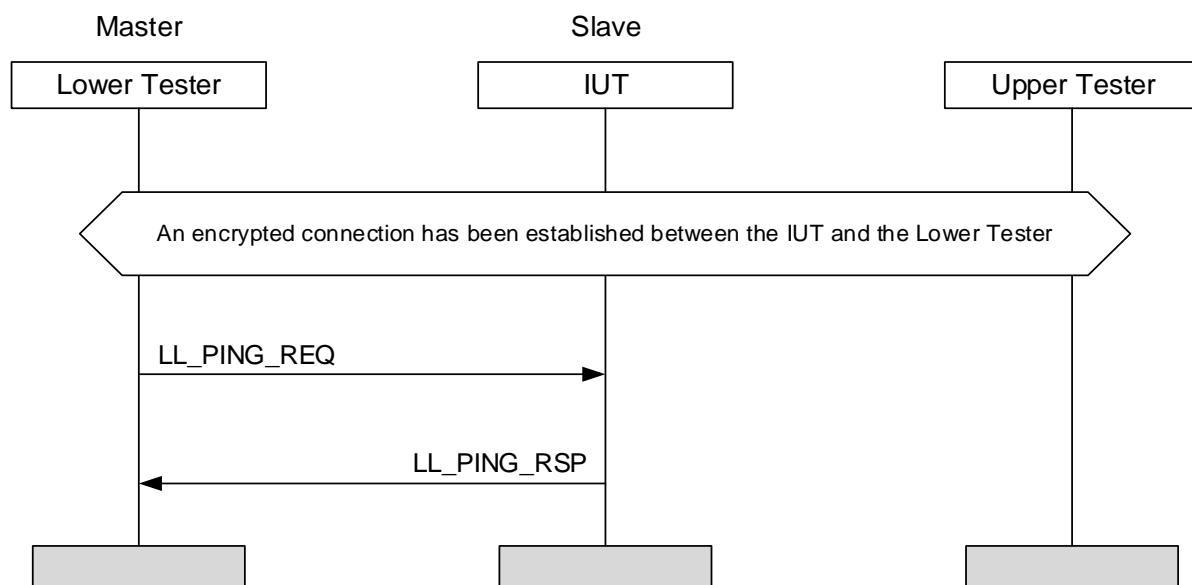


Figure 4.374: LL/SEC/SLA/BV-07-C (Responding to LL_PING_REQ)

1. The Lower Tester transmits the PDU LL_PING_REQ.
 2. The IUT responds to the LL_PING_REQ with an LL_PING_RSP.
- Expected Outcome

Pass Verdict

The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

4.8.5.8 LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ]

- Test Purpose

Verify that the IUT as slave generates the HCI Authenticated Payload Timeout Expired event when the Lower Tester doesn't send a packet containing a valid MIC to the IUT within the Authenticated_Payload_Timeout interval.

- Reference

[3] 5.1.8



- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

- Test Procedure

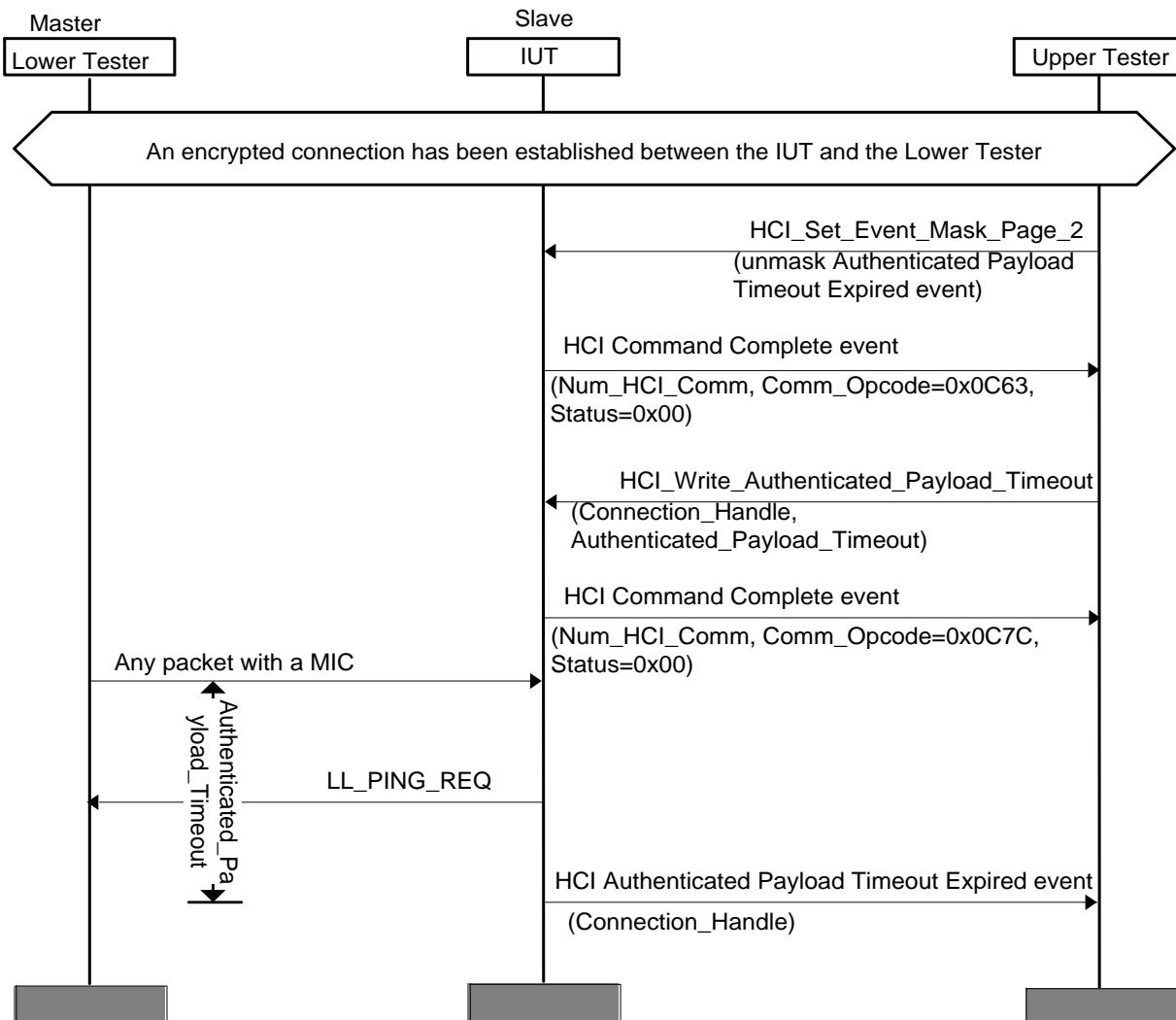


Figure 4.375: LL/SEC/SLA/BV-08-C [No response to LL_PING_REQ]

1. The Upper Tester sets the Authenticated_Payload_Timeout (defined as an IXIT).
2. The Upper Tester unmasks the HCI Authenticated Payload Timeout Expired event.
3. The Lower Tester sends a data packet containing valid MIC.
4. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.
5. The IUT transmits the PDU LL_PING_REQ to the Lower Tester.
6. The Lower Tester does not respond with LL_PING_RSP.
7. The IUT sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester Authenticated_Payload_Timeout (defined as an IXIT) after the last packet that contained a MIC was received by the IUT from the Lower Tester.



- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to the Lower Tester and sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester when the Lower Tester doesn't respond with an LL_PING_RSP.

4.8.5.9 LL/SEC/SLA/BV-09-C [Modified Authentication Payload Timeout]

- Test Purpose

Verify that the IUT as slave uses the correct value of the Authenticated Payload Timeout (greater than 100 s or less than 5 s) set by the Upper Tester.

- Reference

[3] 5.1.8

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

- Test Procedure

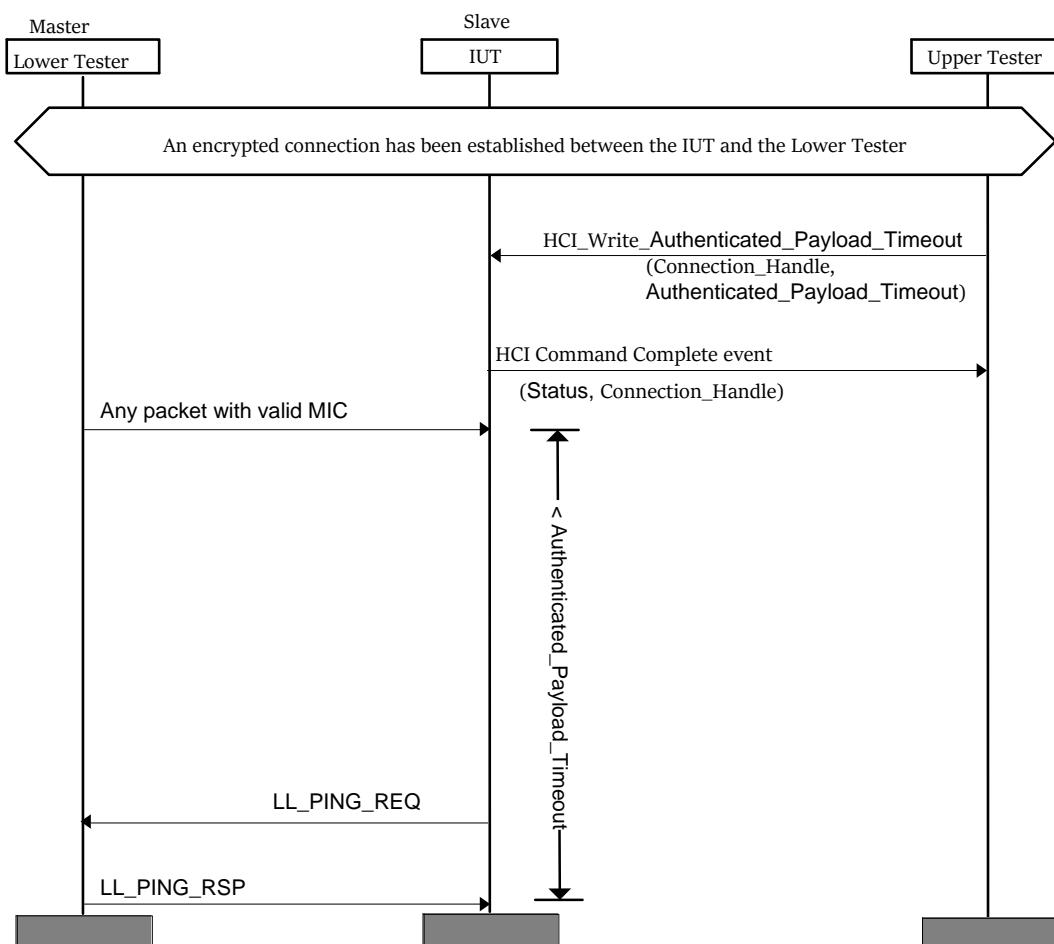


Figure 4.376: LL/SEC/SLA/BV-09-C [Modified Authentication Payload Timeout]



1. The Upper Tester writes Authenticated_Payload_Timeout (defined as an IXIT) less than 5 s.
 2. The Lower Tester sends a data packet containing valid MIC.
 3. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged for time greater than Authenticated_Payload_Timeout (defined as an IXIT).
 4. The IUT transmits the PDU LL_PING_REQ before time which is less than Authenticated_Payload_Timeout (defined as an IXIT).
 5. The Lower Tester responds with LL_PING_RSP. The time between the two packets from the Lower Tester containing a MIC shall not be greater than Authenticated_Payload_Timeout (defined as an IXIT).
- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ and receives the resulting LL_PING_RSP PDU within the time defined in the IXIT for Authenticated_Payload_Timeout after receiving a packet containing a valid MIC from the Lower Tester.

- Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another Authenticated_Payload_Timeout time.

4.8.5.10 LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

- Test Purpose

Verify that the IUT as slave sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_UNKNOWN_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

- Reference

[3] 5.1.8

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

Lower Tester does not support LE Ping Procedure.

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.



- Test Procedure

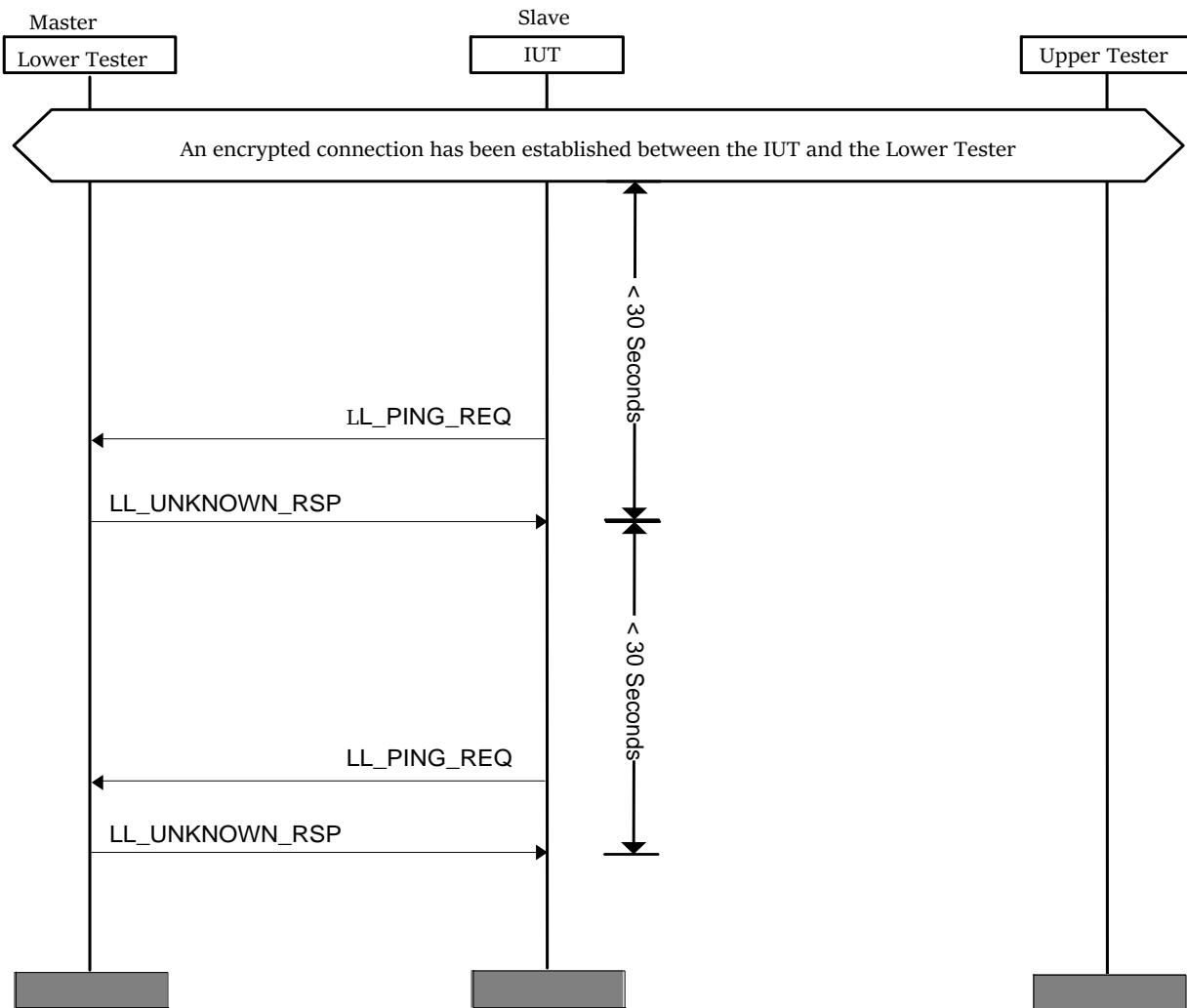


Figure 4.377: LL/SEC/SLA/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_UNKNOWN_RSP.
 2. The Lower Tester responds with LL_UNKNOWN_RSP.
- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.
 - Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.5.11 LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND]

- Test Purpose

Test that, while executing the Encryption Start Procedure, a slave IUT sends a correct LL_REJECT_IND or LL_REJECT_EXT_IND control packet if the Host does not provide a Long Term Key.

The Lower Tester acts as master, maintaining a connection, then initiates the encryption procedure and the Upper Tester fails to provide a Long Term Key to the slave IUT.

- Reference

[3] 5.1.3.1

- Initial Condition

Parameters: LL_slave_connInterval_MIN, LL_slave_connInterval_MAX, LL_slave_connSlaveLatency_MIN, LL_slave_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Slave (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and 'ir').

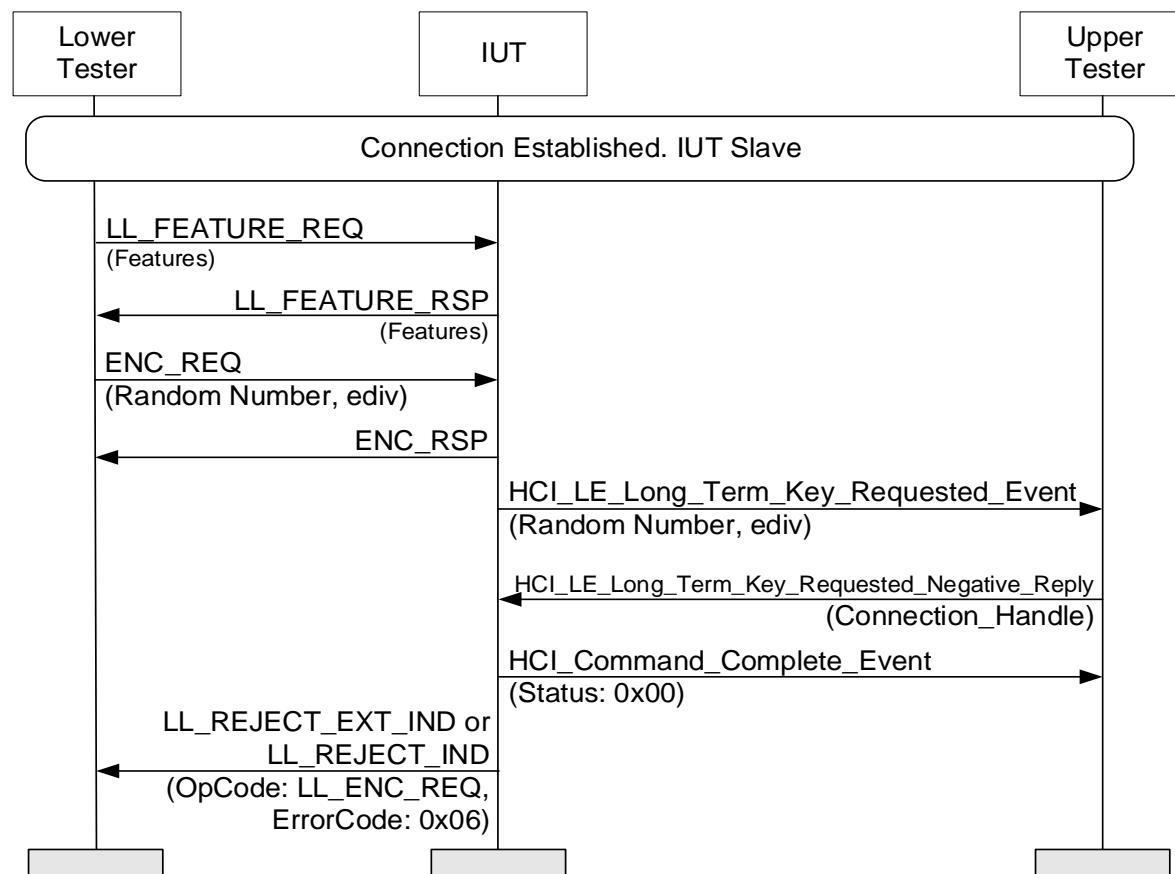


Figure 4.378: LL/SEC/SLA/BV-11-C [Slave Sending LL_REJECT_EXT_IND]



1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT. The Lower Tester initiates a features exchange procedure.
 2. Lower Tester sends an ENC_REQ packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector to the IUT. Expect an ENC_RSP packet in response and continue the master transmissions.
 3. Upper Tester expects an HCI_LE_Long_Term_Key_Requested event from the IUT, containing the random number and 'ediv' parameters sent in step 3.
 4. Upper Tester sends an HCI_LE_Long_Term_Key_Requested_Negative_Reply command to the IUT and expects an HCI_Command_Complete in response.
 5. Lower Tester expects an LL_REJECT_IND or LL_REJECT_EXT_IND PDU from the IUT with the RejectOpcode set to "LL_ENC_REQ" and ErrorCode indicating "PIN or Key missing".
- Expected Outcome

Pass Verdict

Lower Tester receives an ENC_RSP packet in response to the ENC_REQ packet it had previously sent to the IUT.

Upper Tester receives an HCI_Command_Complete event with the Status parameter set to 'Success' in response to the HCI_LE_Long_Term_Key_Requested_Negative_Reply command.

Lower Tester receives an LL_REJECT_IND or LL_REJECT_EXT_IND PDU from the IUT with the RejectOpcode set to "LL_ENC_REQ" and ErrorCode indicating "PIN or Key missing".

4.8.5.12 LL/SEC/SLA/BI-01-C [Slave Encryption Setup: Missing Response]

- Test Purpose

Test that a slave IUT can perform the encryption mode change procedure and recover from a master device failing to send a start encryption response packet.

The Lower Tester acts as a master, maintaining a connection, then initiates the encryption mode setup, failing before the setup is complete.

- Reference

[3] 5.1.3

- Initial Condition

State: Connected Slave (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and "ir").



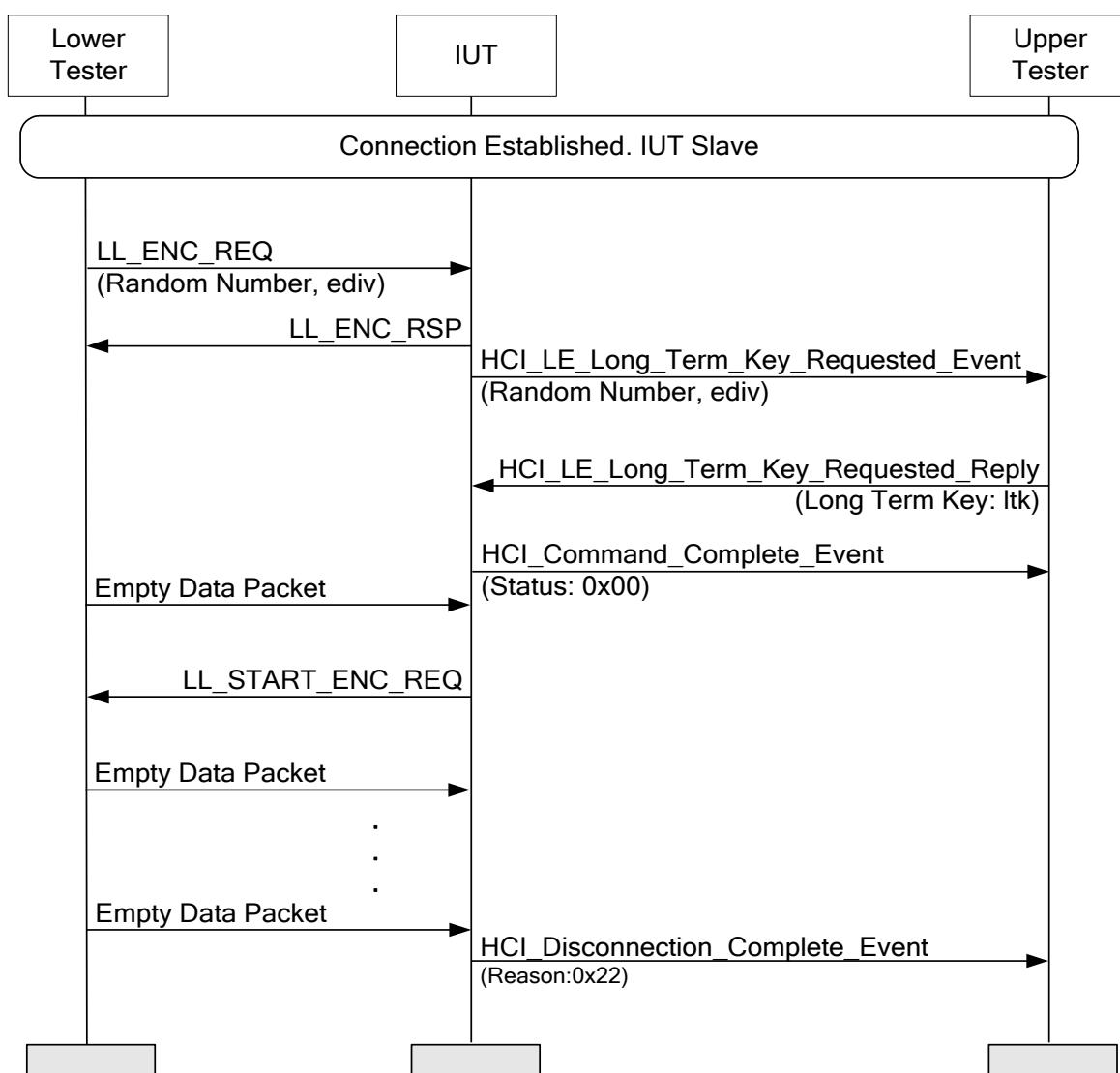


Figure 4.379: LL/SEC/SLA/BI-01-C [Slave Encryption Setup: Missing Response]

1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
2. Lower Tester, calculates the master portions of the session key diversifier, the initialization vector and a random number.
3. Lower Tester sends an **ENC_REQ** packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector to the IUT. Expect a **LL_ENC_RSP** packet in response and continue the master transmissions.
4. Upper Tester expects an **HCI_LE_Long_Term_Key_Requested_Event** from the IUT, containing the random number and 'ediv' parameters sent in step 3.
5. Upper Tester sends an **HCI_LE_Long_Term_Key_Requested_Reply** command with 'ltk' to the IUT and expects an **HCI_Command_Complete** in response.
6. Lower Tester expects an **START_ENC_REQ** packet in response to the master transmissions. Acknowledge the packet.
7. Lower Tester continues the master transmissions, but does not send a **LL_START_ENC_RSP** packet to the IUT.
8. Lower Tester expects the IUT to stop slave transmissions when the connection control timer expires.

9. Interleave with step 8: Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating termination from LL Response Timeout (0x22).

- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping slave transmissions,

The IUT reports the termination of the connection with an HCI event.

4.8.5.13 LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC]

- Test Purpose

Test that a slave IUT terminates an encrypted connection upon a MIC failure in a packet received from the master device.

The Lower Tester acts in the master role, maintaining an encrypted connection and transferring data, then causes connection termination by corrupting the data packet contents.

- Reference

[3] 3.1, 5.1.3

- Initial Condition

State: Encrypted Slave Connection (common identity root, common encryption root, any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

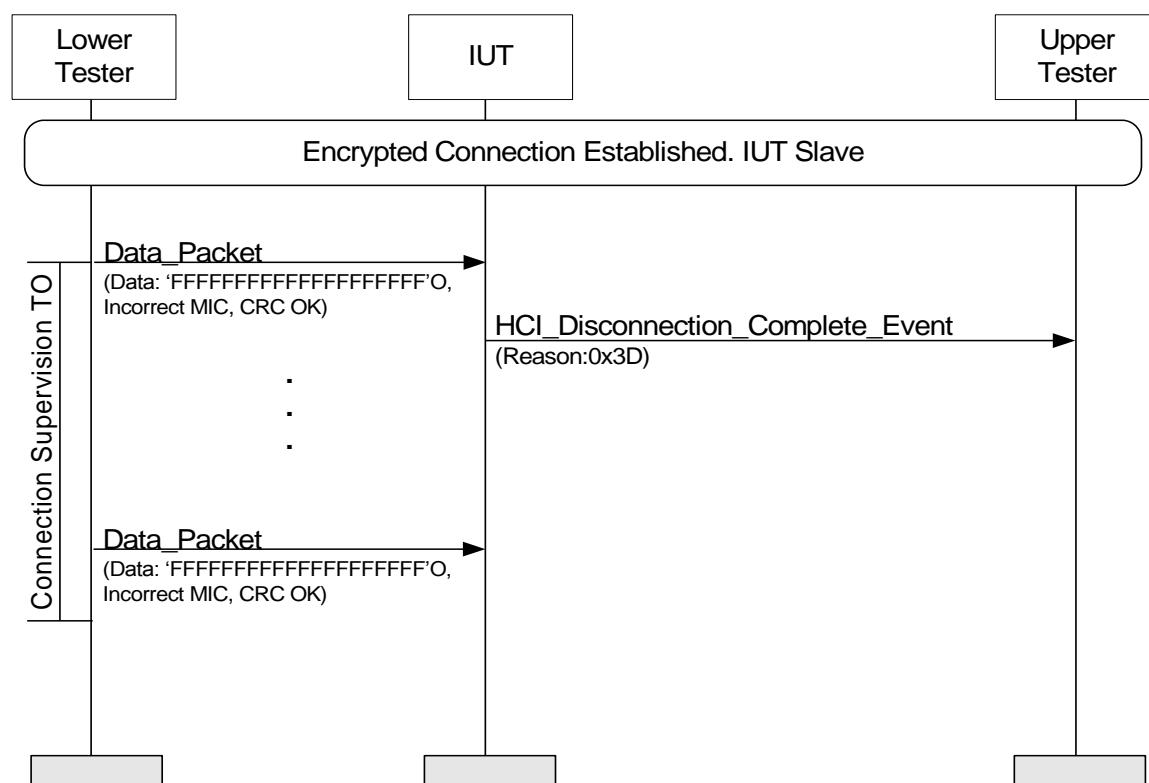


Figure 4.380: LL/SEC/SLA/BI-03-C [Slave MIC Failure: Corrupted MIC]



1. Preamble: starting from an encrypted connection setup using the common connection parameters and encryption key variables
 2. Lower Tester sends a DATA packet with 10 bytes of data, flipping a single bit in the MIC after calculation, but using a correct CRC for the invalid encrypted packet.
 3. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure
 4. Repeat step 2 expecting no response from the IUT up to a time equal to the connection supervision timeout value.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT terminating the connection after having tried to authenticate a corrupted data packet.

The IUT does not report to the Upper Tester any of the corrupted data packets received from the Lower Tester.

The IUT reports the connection termination with an HCI event.

4.8.5.14 LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header]

- Test Purpose

Test that a slave IUT terminates an encrypted connection upon a MIC failure in a packet received from the master device.

The Lower Tester acts in the master role, maintaining an encrypted connection and transferring data, then causes connection termination by corrupting the data packet contents.

- Reference

[3] 3.1, 5.1.3

- Initial Condition

State: Encrypted Slave Connection (common identity root, common encryption root, any advertising interval, any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.



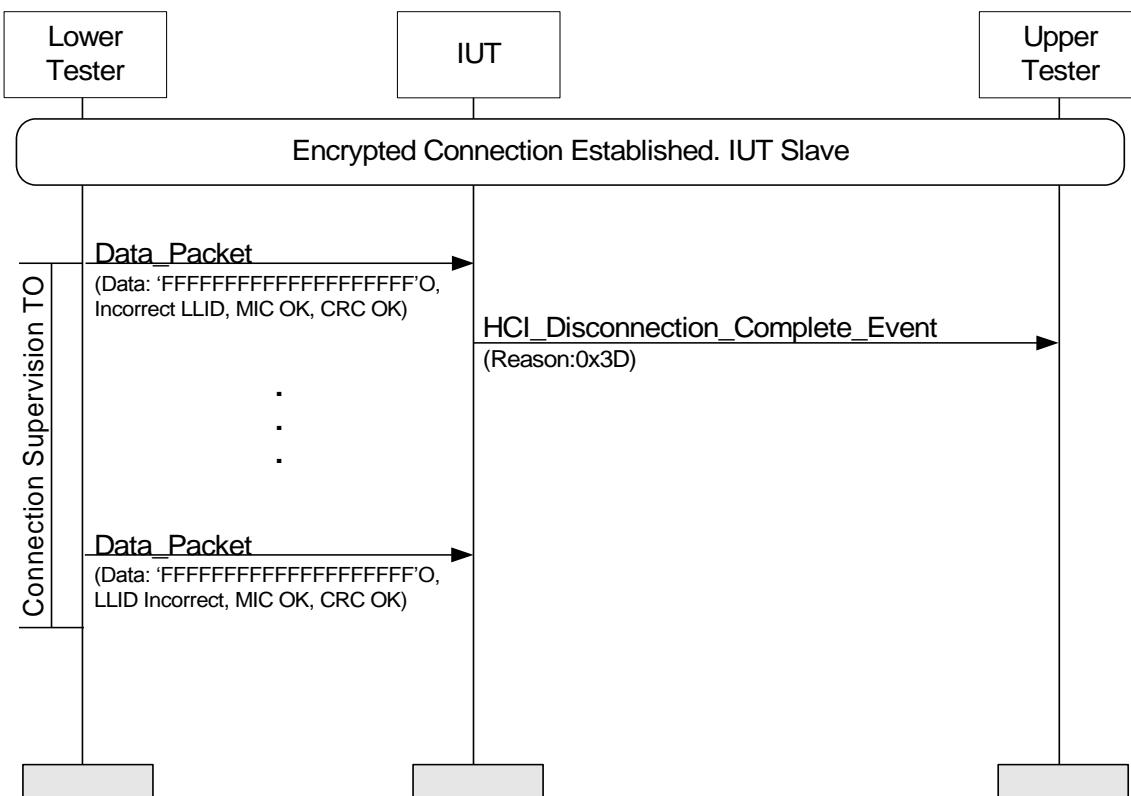


Figure 4.381: LL/SEC/SLA/BI-04-C [Slave MIC Failure: Corrupted Header]

1. Lower Tester sends a DATA packet with 10 bytes of data, flipping a single bit in the header (LLID or RFU) after MIC calculation, but using a correct CRC for the invalid encrypted packet.
 2. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
 3. Repeat step 1 expecting no response from the IUT, up to a time equal to the connection supervision timeout value.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT terminating the connection after having tried to authenticate a corrupted data packet received from the Lower Tester.

The IUT does not report to the Upper Tester any of the corrupted data packets received from the Lower Tester.

The IUT reports the connection termination with an HCI event.

4.8.5.15 LL/SEC/SLA/BI-05-C [Slave Receiving unexpected PDU during encryption start]

- Test Purpose

Test that a slave IUT which has started the encryption procedure does not respond to an LL_VERSION_IND but instead drops the link.

The Lower Tester acts as a master.

- Reference

[3] 5.1.3.1



- Initial Condition

State: Connected Slave (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value)

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and “ir”).

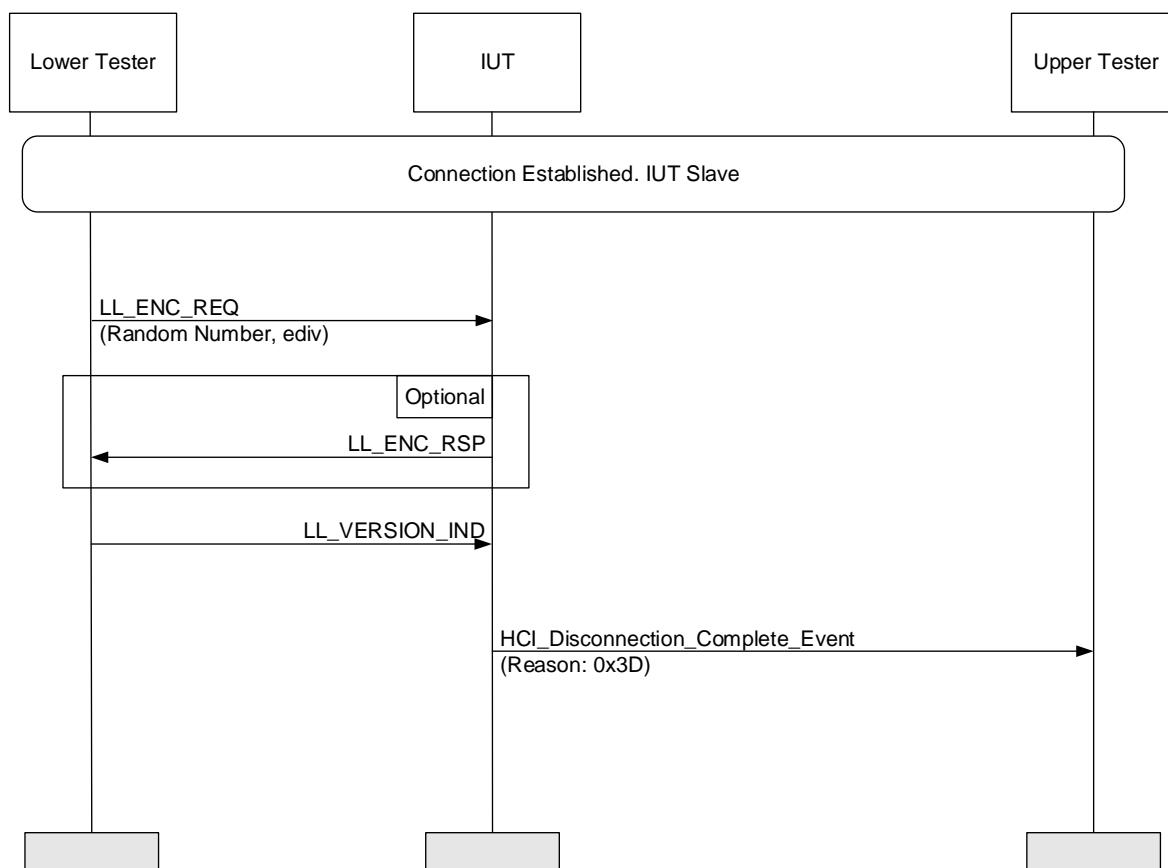


Figure 4.382: LL/SEC/SLA/BI-05-C [Slave Receiving unexpected Data Channel PDU during encryption start]

- In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
- Lower Tester calculates the master portions of the session key diversifier, the initialization vector and a random number.
- Lower Tester sends an LL_ENC_REQ packet, containing the random number, ‘ediv’ and master portions of the session key diversifier and the initialization vector to the IUT.
- The LL_ENC_REQ packet is immediately followed by a LL_VERSION_IND packet from the Lower Tester. (In the same connection event if possible).
- [Optional] IUT may send LL_ENC_RSP
- The IUT should terminate the connection on receiving the LL_VERSION_IND packet.
- Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination from “Connection Terminated Due to MIC Failure (0x3D).”

- Expected Outcome

Pass verdict:

The IUT terminates the connection upon receiving the unexpected LL_VERSION_IND packet.



4.8.6 MAS

Tests that the IUT behaves according to the security procedures in the master role.

4.8.6.1 LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup]

- **Test Purpose**

Test that a master IUT is able to request encryption mode setup from a slave Controller, correctly encrypting the packets and performing the procedure.

The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT.

- **Reference**

[3] 5.1.3

- **Initial Condition**

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout)

- **Test Procedure**

Execute the test procedure with the common connection parameters and using the common variables for the encryption roots (variables 'er', and 'ir').



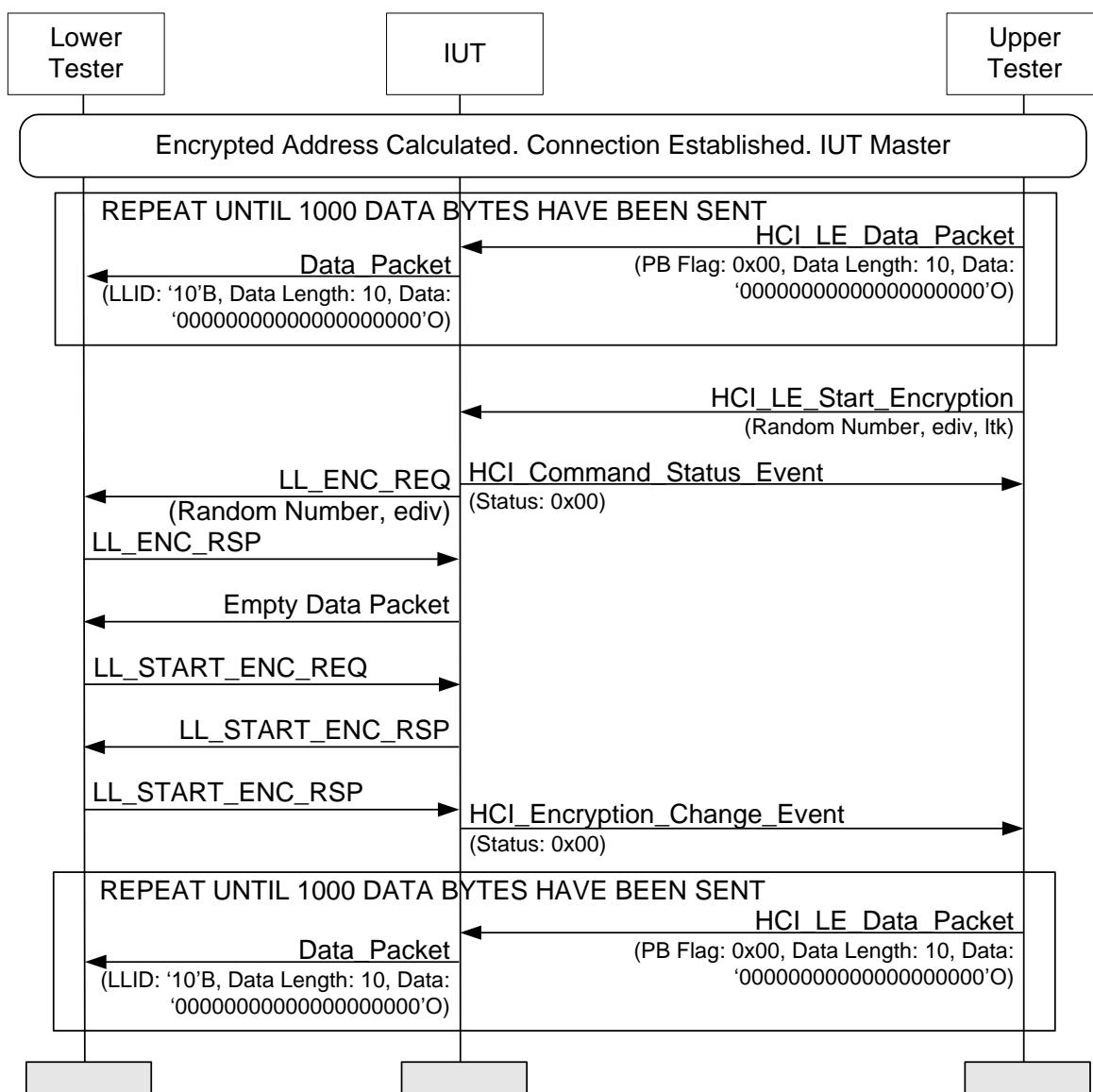


Figure 4.383: LL/SEC/MAS/BV-01-C [Master Encryption Mode Setup]

1. In the preamble steps: connection establishment use public address type for the Lower Tester and a supported type of address for the IUT.
2. Configure Upper Tester to submit data elements to the IUT with the HCI LE Data Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
3. Lower Tester expects DATA packets from the IUT, including the data sent in step 2 and sends empty DATA packets to the IUT until all data sent in 2 has been received.
4. Lower Tester, calculates the slave portions of the session key diversifier, the initialization vector and a random number
5. Upper Tester sends an HCI LE Start Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution and 'Itk'. Expect an HCI_Command_Status event in response.
6. Lower Tester expects an ENC_REQ packet including the parameters from 5: with the master portions of the session key diversifier and the initialization vector. Acknowledge the encryption

- request and respond with an ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.
7. Configure the Lower Tester calculating the session key diversifier and initialization vector from the parameters exchanged in step 6. Calculate the diversifier, using 'dhk' and the parameters received in step 6. Calculate the 'ltk' from the diversifier using 'er'. Check the ltk to match the calculations in the preamble steps. Calculate the session key using the session key diversifier.
 8. Lower Tester expects an empty DATA packet from the IUT and responds with an START_ENC_REQ packet.
 9. Lower Tester expects an START_ENC_RSP packet encrypted from the IUT and responds with an START_ENC_RSP packet encrypted.
 10. Upper Tester expects an HCI_Encryption_Change event from the IUT, containing the connection handle from the preamble steps' execution.
 11. Configure Upper Tester to submit data elements to the IUT with the HCI_LE_Data_Packet command using the connection handle, including the Packet_Boundary_Flag flag set and data elements with the value 0x00, for a data total length of 10, until the selected number of octets (1000) are successfully submitted.
 12. Lower Tester expects the DATA packets received from the IUT to be encrypted with matching MIC fields and sends empty DATA packets in response until all data sent in step 2 have been received.
 13. Master Connection Terminated (connection interval, slave latency, timeout, channel map, unencrypted, connection handle).
 14. In the preamble steps (connection establishment use random address type for the Lower Tester and a random type of address for the IUT).
 15. Repeat steps 2-13.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the data transmitted correctly reported by the IUT.

The IUT sends its initialization vector and session key diversifier in an ENC_REQ packet.

The IUT sends a START_ENC_RSP packet until acknowledged.

The IUT reports the encryption mode setup with the HCI event.

4.8.6.2 LL/SEC/MAS/BV-02-C [Master Pause Encryption]

- Test Purpose

Test that a master IUT can perform the encryption pause procedure.

The Lower Tester acts in the slave role, accepts the encryption pause request from the IUT.

- Reference

[3] 5.1.3.2

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout)



- Test Procedure

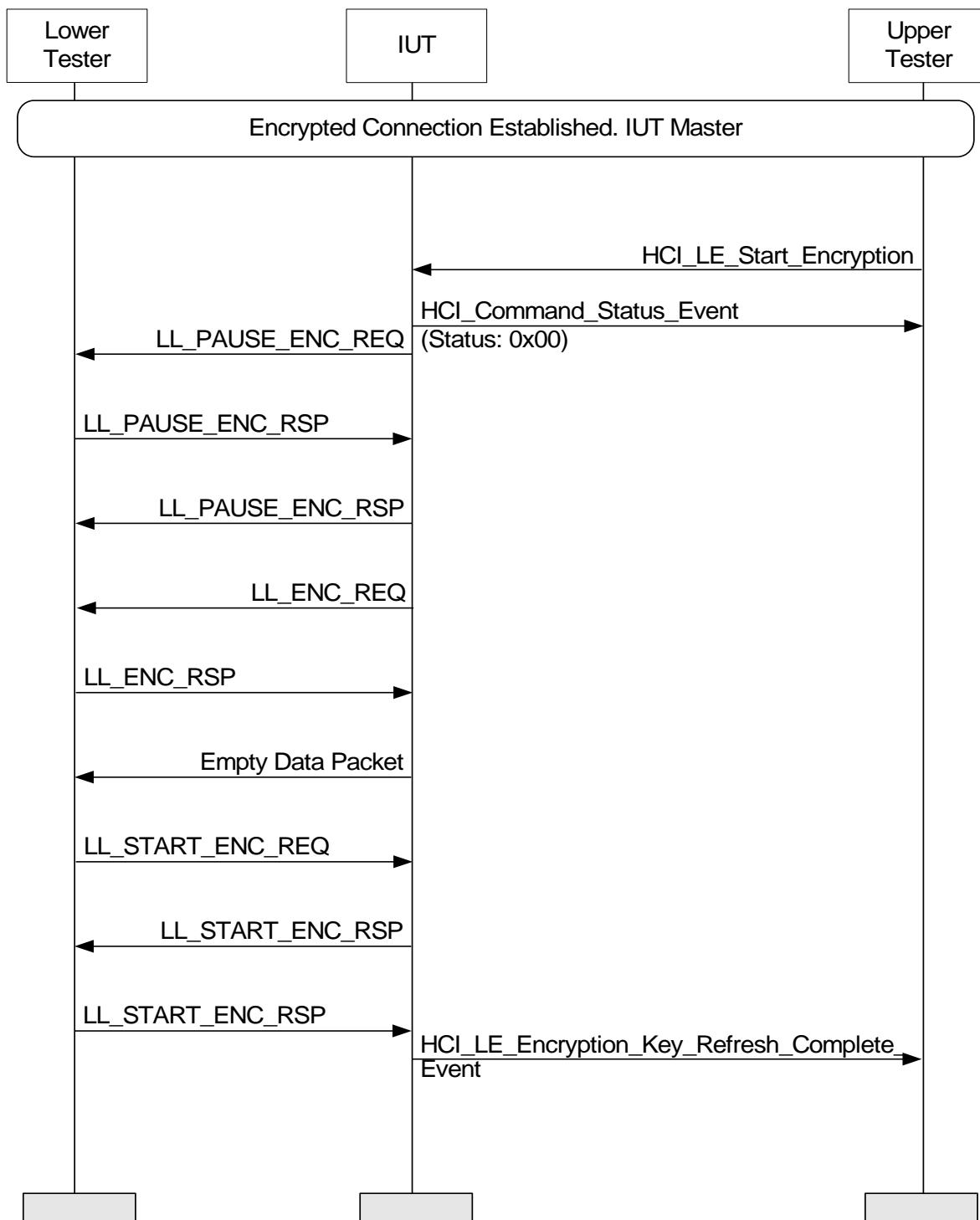


Figure 4.384: LL/SEC/MAS/BV-02-C [Master Pause Encryption]

- As preamble: an encrypted connection is established (IUT acts as master)
- Upper Tester sends an HCI_LE_Start_Encryption and expects an HCI_Command_Status in response.
- Lower Tester expects an encrypted PAUSE_ENC_REQ packet and sends an encrypted PAUSE_ENC_RSP packet in response.



4. Lower Tester expects an unencrypted PAUSE_ENC_RSP packet from the IUT. At this point the connection stops being encrypted.
 5. Lower Tester expects an ENC_REQ packet to re-enable encryption using a new session key and send an ENC_RSP packet in response.
 6. Lower Tester sends an START_ENC_REQ packet in response to the master transmissions and expects the acknowledgement of the packet.
 7. Lower Tester expects an START_ENC_RSP packet encrypted from the IUT and sends a START_ENC_RSP packet encrypted in response.
 8. Upper Tester expects an HCI_Encryption_Key_Refresh event.
 9. Master Connection Terminated (connection interval, slave latency, timeout, channel map, un-encrypted, connection handle).
- Expected Outcome

Pass Verdict

The IUT starts the encryption pause procedure upon reception of HCI_LE_Start_Encryption.

The IUT sends the ENC_REQ packet.

The IUT sends the event HCI_Encryption_Key_Refresh_Complete_Event once encryption is resumed.

4.8.6.3 LL/SEC/MAS/BV-03-C [Master Receiving Reject_Ind]

- Test Purpose

Test that, while executing the Encryption Start Procedure, a master IUT receives a Reject_Ind control packet and notifies it to the Host.

The Lower Tester acts as slave, maintaining a connection, then sends a Reject_Ind packet to the master IUT.

- Reference

[3] 5.1.3.1

- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, private address, any advertising channel map, common connection interval, common timeout, any SCA value).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and "ir").



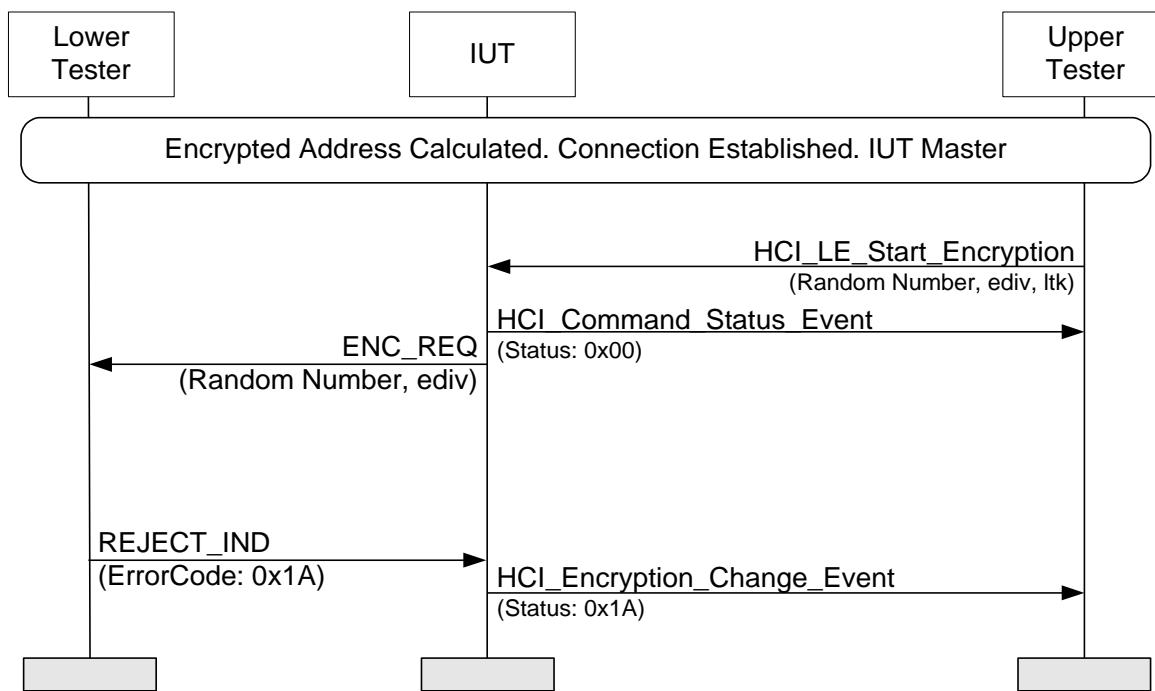


Figure 4.385: LL/SEC/MAS/BV-03-C [Master Receiving Reject_Ind]

1. In the preamble steps (connection establishment) use public address type for the Lower Tester and a supported type of address for the IUT.
 2. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT and expects an HCI_Command_Status event in response.
 3. Lower Tester receives an ENC_REQ packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector from the IUT.
 4. Lower Tester sends a REJECT_IND packet to the IUT with ErrorCode indicating "Unsupported Remote Feature / Unsupported LMP Feature".
 5. Upper Tester expects an HCI_Encryption_Change event from the IUT with error code indicating "Unsupported Remote Feature / Unsupported LMP Feature".
- Expected Outcome

Pass Verdict

Lower Tester receives an ENC_REQ packet, containing the random number 'ediv', from the IUT.

Upper Tester receives an HCI_Encryption Change event with the 'Status' parameter set to 'Unsupported Remote Feature / Unsupported LMP Feature' from the IUT.

4.8.6.4 LL/SEC/MAS/BV-04-C [Master Encryption: Sending Data before LL_ENC_RSP]

- Test Purpose

Test, during the encryption start procedure, that a master IUT reports data received after master sends LL_ENC_REQ and before master receives LL_ENC_RSP and does not terminate the established connection.

The Lower Tester acts as in the slave role, first receiving the encryption mode setup request from the IUT, then sends a data packet before sending LL_ENC_RSP.

- Reference

[3] 5.1.3



- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).

- Test Procedure

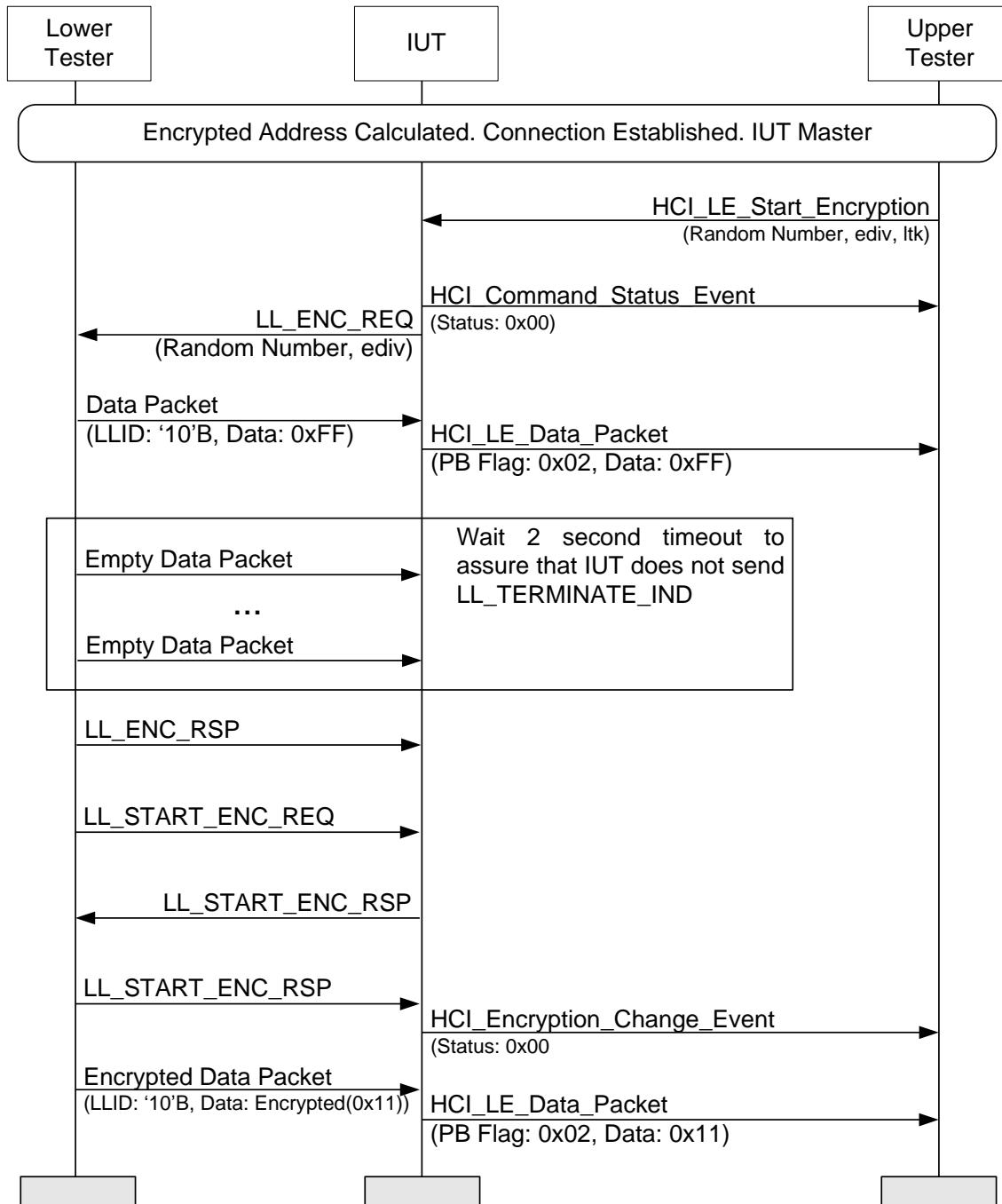


Figure 4.386: LL/SEC/MAS/BV-04-C [Master Encryption: Sending Data before LL_ENC_RSP]

- Upper Tester sends an **HCI_LE_Start_Encryption** command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an **HCI_Command_Status** event in response.



2. Lower Tester expects a LL_ENC_REQ packet including the parameters from step 1.
3. Lower Tester sends a DATA packet with 0xFF as data
4. Upper Tester expects an HCI_LE_Data_Packet with 0xFF as data
5. Lower Tester sends an LL_ENC_RSP packet.
6. Lower Tester sends an LL_START_ENC_REQ packet.
7. Lower Tester expects an LL_START_ENC_RSP packet encrypted from the IUT and responds with an LL_START_ENC_RSP packet encrypted.
8. Upper Tester expects an HCI_Encryption_Change event from the IUT, containing the connection handle from the preamble steps' execution.
9. Lower Tester sends a data packet with encrypted (0x11) as data.
10. Upper Tester expects an HCI_LE_Data_Packet with 0x11 as data.

- Expected Outcome

Pass Verdict

The IUT does not send a LL_TERMINATE.

The IUT sends an HCI_LE_Data_Packet, with data 0xFF, before sending HCI_Encryption_Change event.

The IUT sends an HCI_Encryption_Change event with status set to "SUCCESS" (0x00)

The IUT sends an HCI_LE_Data_Packet, with data 0x11 after sending HCI_Encryption_Change event.

The IUT does not send an HCI_Disconnection_Complete event

4.8.6.5 LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]

- Test Purpose

Tests, during the encryption resume procedure, that a master IUT reports data received after master sends LL_PAUSE_ENC_REQ and before master receives LL_PAUSE_ENC_RSP and does not terminate the established connection.

The Lower Tester acts as in the slave role, first accepting the encryption pause request from the IUT, then sends a data packet before sending LL_PAUSE_ENC_RSP.

- Reference

[3] 5.1.3.2

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).



- Test Procedure

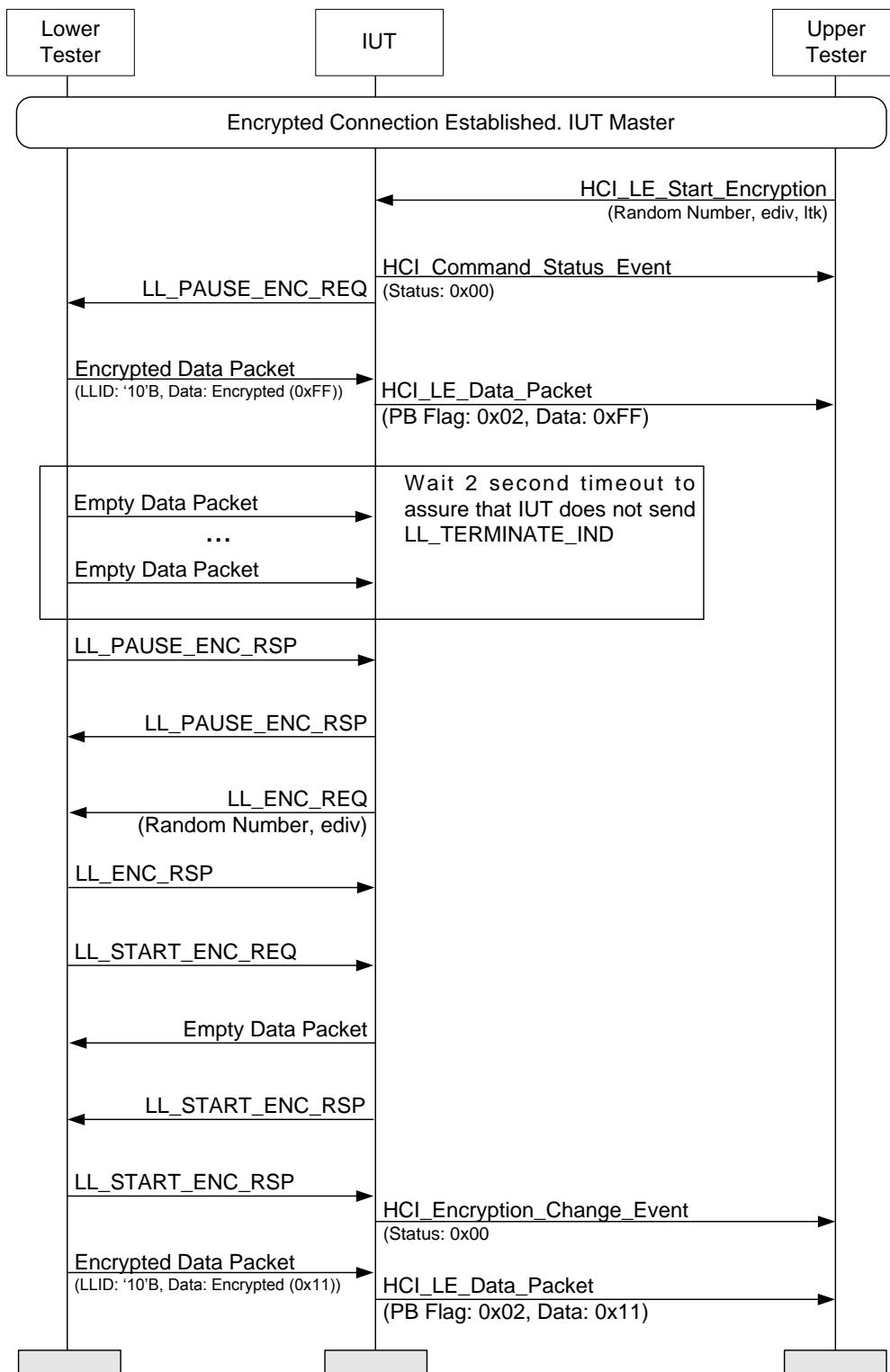


Figure 4.387: LL/SEC/MAS/BV-05-C [Master Pause Encryption: Sending Data before LL_PAUSE_ENC_RSP]

- Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.
- Lower Tester expects an encrypted PAUSE_ENC_REQ packet.



3. Lower Tester sends a data packet with encrypted (0xFF) as data.
 4. Upper Tester expects an HCI_LE_Data_Packet with 0xFF as data.
 5. Lower Tester sends an encrypted PAUSE_ENC_RSP packet.
 6. Lower Tester expects an unencrypted PAUSE_ENC_RSP packet from the IUT. At this point the connection stops being encrypted.
 7. Lower Tester expects a LL_ENC_REQ packet to re-enable encryption using a new session key, and sends an LL_ENC_RSP packet in response.
 8. Lower Tester sends an LL_START_ENC_REQ packet.
 9. Lower Tester expects an LL_START_ENC_RSP packet encrypted from the IUT and responds with an LL_START_ENC_RSP packet encrypted.
 10. Upper Tester expects an HCI_Encryption_Key_Refresh event from the IUT.
 11. Lower Tester sends a data packet with encrypted (0x11) as data.
 12. Upper Tester expects an HCI_LE_Data_Packet with 0x11 as data.
- Expected Outcome

Pass Verdict

The IUT does not send a LL_TERMINATE_IND.

The IUT sends an HCI_LE_Data_Packet, with data 0xFF, before sending HCI_Encryption_Change event.

The IUT sends an HCI_Encryption_Key_Refresh event with status set to "SUCCESS" (0x00)

The IUT sends an HCI_LE_Data_Packet, with data 0x11 after sending HCI_Encryption_Change event.

The IUT does not send an HCI_Disconnection_Complete event.

4.8.6.6 LL/SEC/MAS/BV-06-C [Initiate LE Ping procedure when encryption is enabled]

- Test Purpose

Verify that the IUT as master sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_PING_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

- Reference

[3] 5.1.8

- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.



- Test Procedure

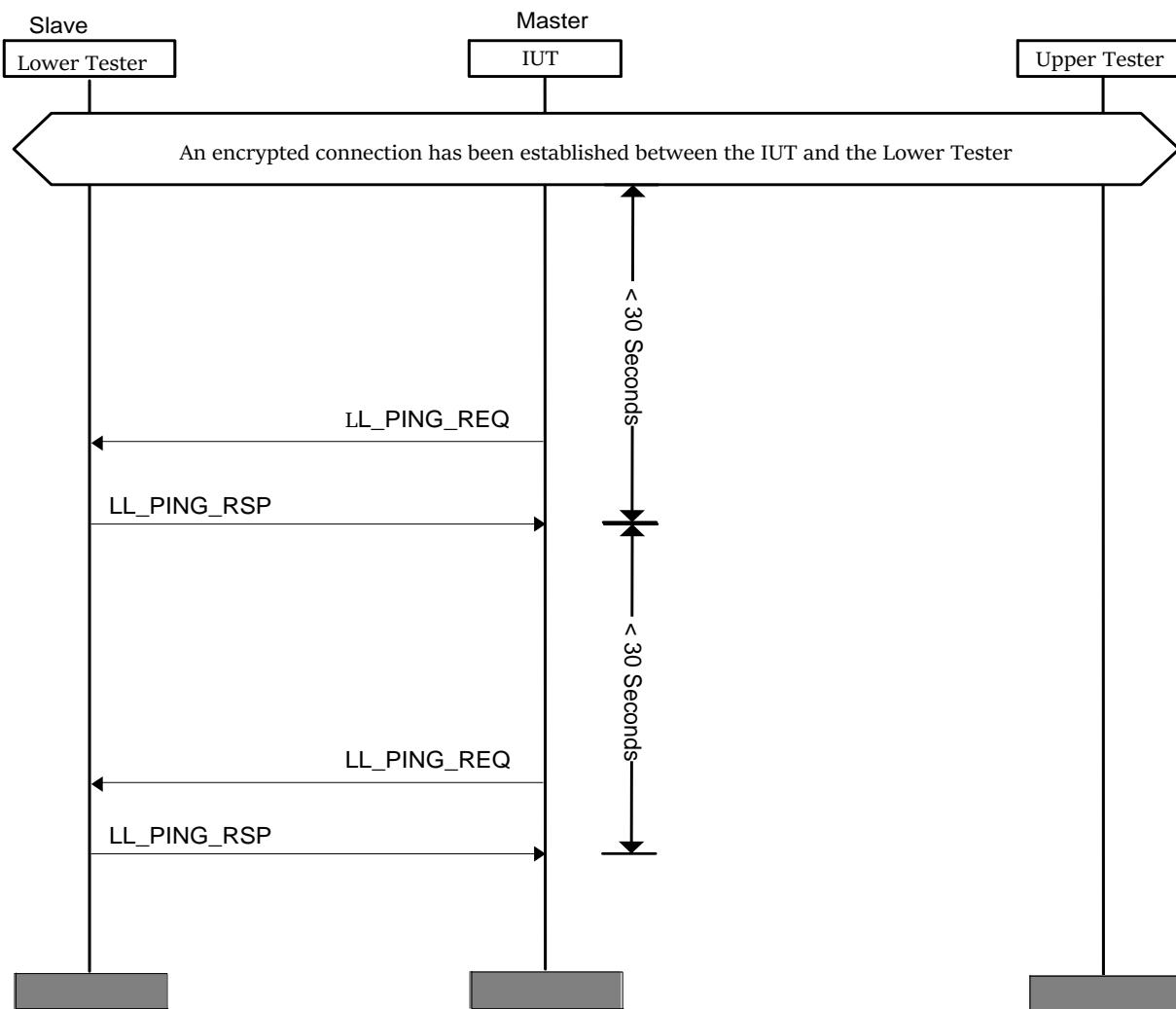


Figure 4.388: LL/SEC/MAS/BV-06-C (Initiate LE Ping procedure when encryption is enabled)

1. The IUT transmits the PDU LL_PING_REQ less than 30 s after receiving a LL_PING_RSP.
2. The Lower Tester responds with LL_PING_RSP.

- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to trigger a LL_PING_RSP over the air before the 30 s Payload Authentication Timeout expires.

- Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.6.7 LL/SEC/MAS/BV-07-C [Responding to LL_PING_REQ]

- Test Purpose

Verify that the IUT as master responds to an LL_PING_REQ sent by the Lower Tester.



- Reference

[3] 5.1.8

- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged

- Test Procedure

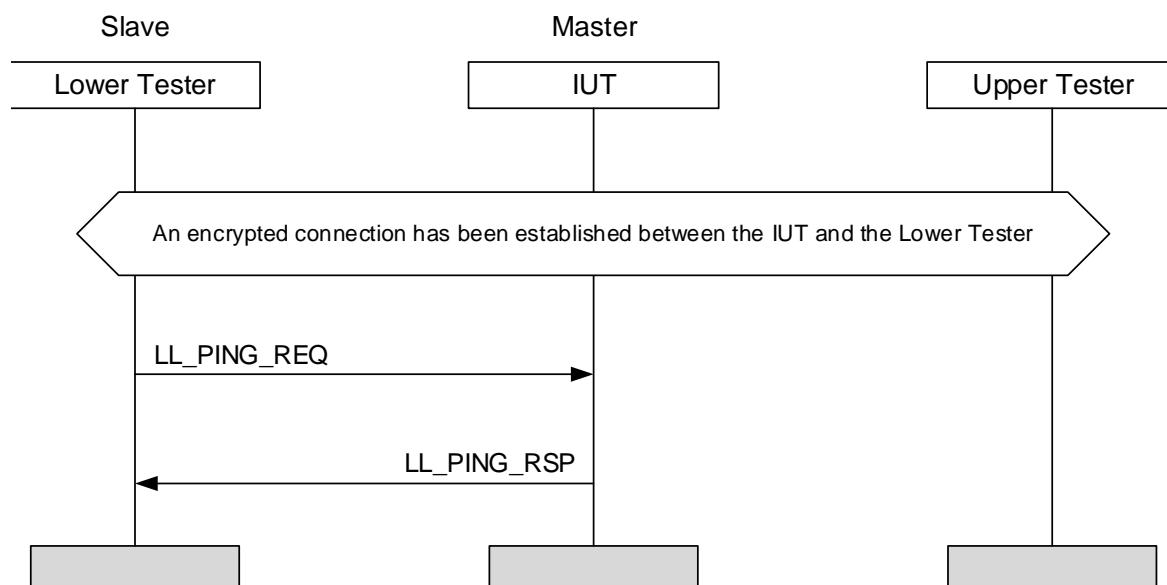


Figure 4.389: LL/SEC/MAS/BV-07-C (Responding to LL_PING_REQ)

1. The Lower Tester transmits the PDU LL_PING_REQ.
2. The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

- Expected Outcome

Pass Verdict

The IUT responds to the LL_PING_REQ with an LL_PING_RSP.

4.8.6.8 LL/SEC/MAS/BV-08-C [No response to LL_PING_REQ]

- Test Purpose

Verify that the IUT as master generates the HCI Authenticated Payload Timeout Expired event when the Lower Tester doesn't send a packet containing a valid MIC to the IUT within the Authenticated_Payload_Timeout interval.

- Reference

[3] 5.1.8



- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

The connection is encrypted.

- Test Procedure

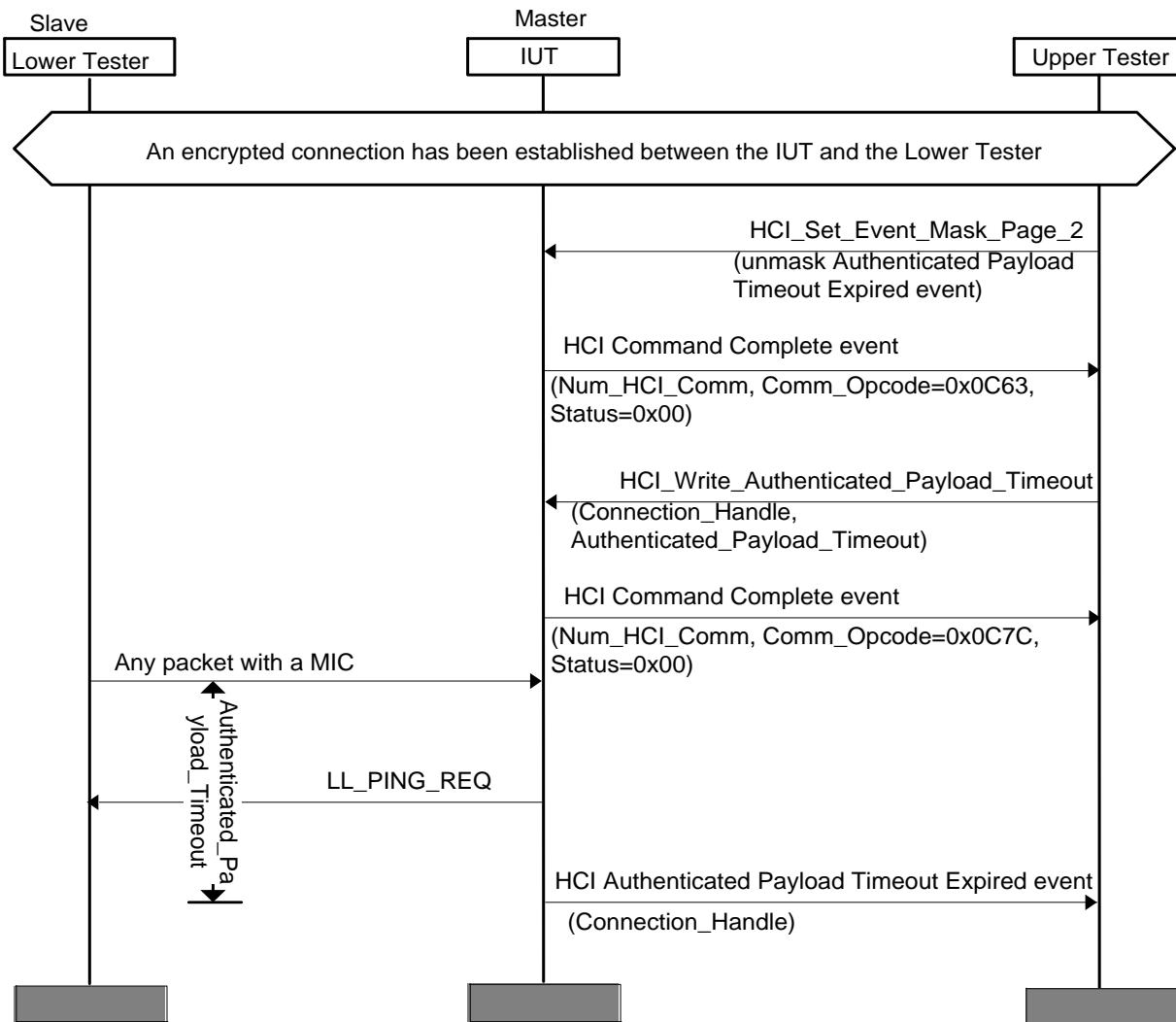


Figure 4.390: LL/SEC/MAS/BV-08-C (No response to LL_PING_REQ)

1. The Upper Tester sets the Authenticated_Payload_Timeout (defined as an IXIT).
2. The Upper Tester unmasks the HCI Authenticated Payload Timeout Expired event.
3. The Lower Tester sends a data packet containing valid MIC.
4. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.
5. The IUT transmits the PDU LL_PING_REQ to the Lower Tester.
6. The Lower Tester does not respond with LL_PING_RSP.
7. The IUT sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester Authenticated_Payload_Timeout (defined as an IXIT) after the last packet that contained a MIC was received by the IUT from the Lower Tester.

- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ to the Lower Tester and sends an HCI Authenticated Payload Timeout Expired event to the Upper Tester when the Lower Tester doesn't respond with an LL_PING_RSP.

4.8.6.9 LL/SEC/MAS/BV-09-C [Modified Authentication Payload Timeout]

- Test Purpose

Verify that the IUT as master uses the correct value of the Authenticated Payload Timeout (greater than 100 s or less than 5 s) set by the Upper Tester.

- Reference

[3] 5.1.8

- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

- Test Procedure

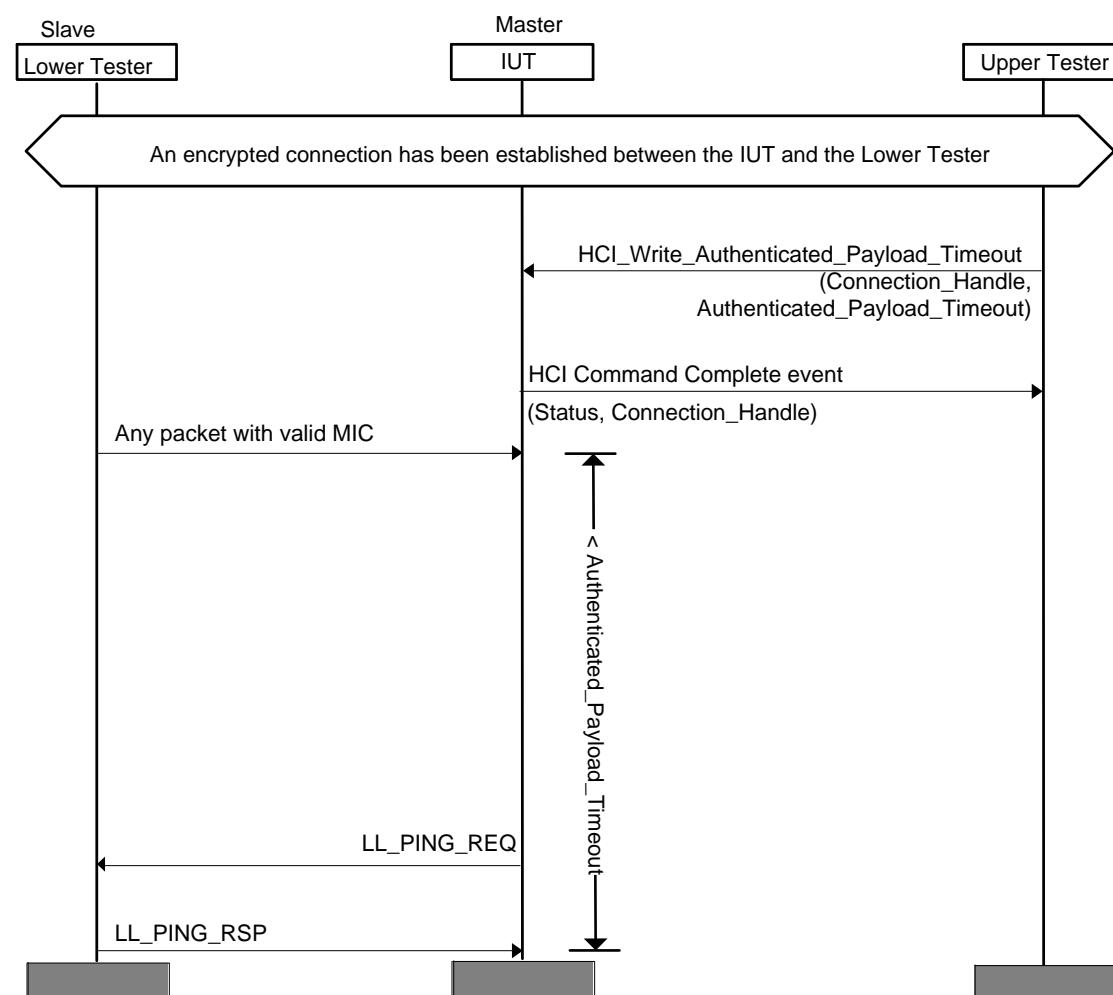


Figure 4.391: LL/SEC/MAS/BV-09-C (Modified Authentication Payload Timeout)



1. The Upper Tester writes Authenticated_Payload_Timeout (defined as an IXIT) less than 5 s.
 2. The Lower Tester sends a data packet containing valid MIC.
 3. The LE ACL connection is kept idle i.e. no LE-U or LE-C traffic is exchanged for time greater than Authenticated_Payload_Timeout (defined as an IXIT).
 4. The IUT transmits the PDU LL_PING_REQ before time which is less than Authenticated_Payload_Timeout (defined as an IXIT).
 5. The Lower Tester responds with LL_PING_RSP. The time between the two packets from the Lower Tester containing a MIC shall not be greater than Authenticated_Payload_Timeout (defined as an IXIT).
- Expected Outcome

Pass Verdict

The IUT transmits the PDU LL_PING_REQ and receives the resulting LL_PING_RSP PDU within the time defined in the IXIT for Authenticated_Payload_Timeout after receiving a packet containing a valid MIC from the Lower Tester.

- Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another Authenticated_Payload_Timeout time.

4.8.6.10 LL/SEC/MAS/BV-10-C [Initiate LE Ping procedure when the other side does not support the procedure]

- Test Purpose

Verify that the IUT as master sends an LL_PING_REQ, when a packet containing valid MIC is not received from the Lower Tester for time less than default value of LE Authenticated Payload Timeout, in order to force the Lower Tester to transmit an LE ACL packet (LL_UNKNOWN_RSP). IUT has LE Authenticated Payload Timeout Timer set to default value of 30 s.

The Lower Tester acts in the master role in a maintained connection and responds to the request from the IUT to combat forged acknowledgements.

- Reference

[3] 5.1.8

- Initial Condition

State: Connected Master (any advertising interval, any advertising interval, supported type of own address, any advertising channel map, common connection interval, up to LL_slave_connSlaveLatency_MAX, selected timeout, any SCA value)

Lower Tester does not support LE Ping Procedure.

The connection is encrypted.

The connection is kept idle i.e. no LE-U or LE-C traffic is exchanged.



- Test Procedure

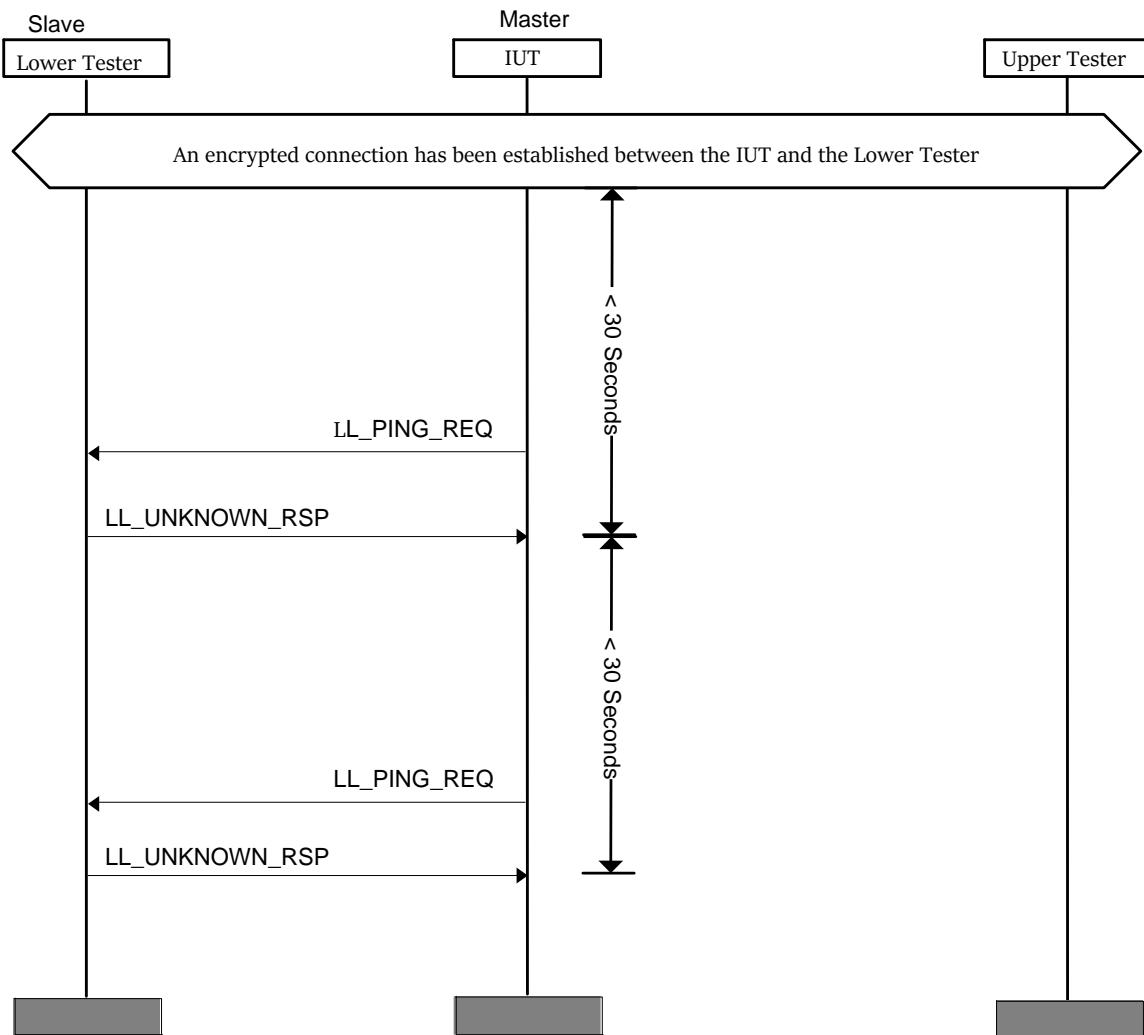


Figure 4.392: LL/SEC/MAS/BV-10-C (Initiate LE Ping procedure when the other side does not support the procedure)

1. The IUT transmits the PDU **LL_PING_REQ** less than 30 s after the establishment of an encrypted connection.
 2. The Lower Tester responds with **LL_UNKNOWN_RSP**.
 3. The IUT transmits the PDU **LL_PING_REQ** less than 30 s after receiving the **LL_UNKNOWN_RSP**.
 4. The Lower Tester responds with **LL_UNKNOWN_RSP**.
- Expected Outcome

Pass Verdict

The IUT transmits the **LL_PING_REQ** PDU to trigger an **LL_UNKNOWN_RSP** PDU over the air before the 30 s Payload Authentication Timeout expires.



- Notes

The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 s.

4.8.6.11 LL/SEC/MAS/BV-11-C [Master Receiving LL_REJECT_EXT_IND]

- Test Purpose

Tests that, while executing the Encryption Start Procedure, a master IUT receives an LL_REJECT_EXT_IND control packet and notifies it to the Host.

The Lower Tester acts as slave, maintaining a connection, then sends an LL_REJECT_EXT_IND packet to the master IUT.

- Reference

[3] 5.1.3.1

- Initial Condition

Parameters: LL_master_connInterval_MIN, LL_master_connInterval_MAX, LL_master_connSlaveLatency_MIN, LL_master_connSlaveLatency_MAX, LL_connTimeout_MIN, LL_connTimeout_MAX.

State: Connected Master (any scan interval, any scan window, public peer address, Lower Tester address, supported type of own address, connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and 'ir').

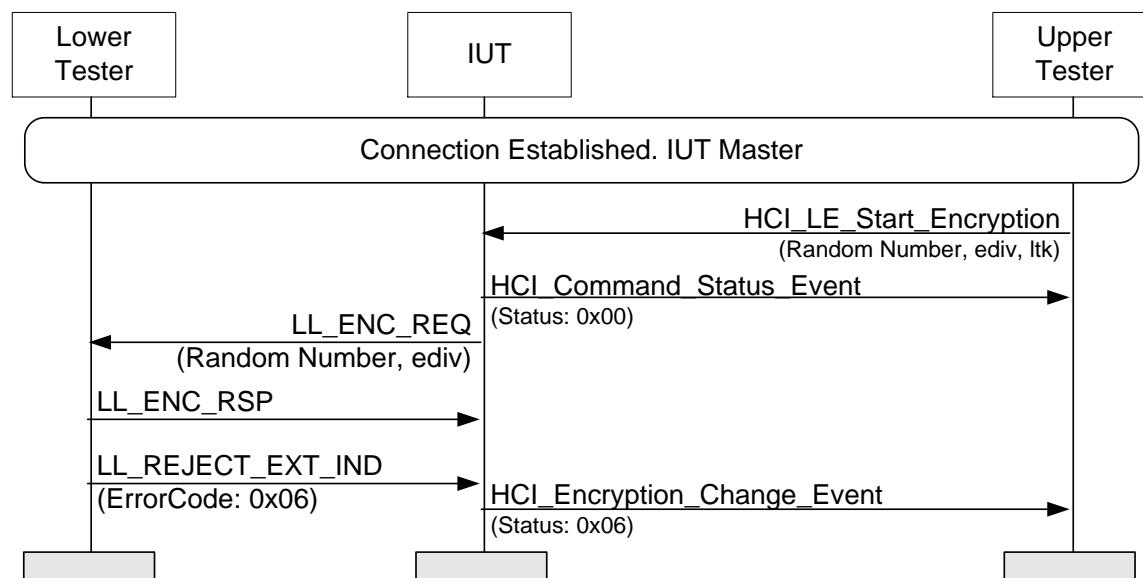


Figure 4.393: LL/SEC/MAS/BV-11-C [Master receiving LL_REJECT_EXT_IND]



1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT and expects an HCI_Command_Status event in response.
 2. Lower Tester receives an LL_ENC_REQ packet, containing the random number, 'ediv' and master portions of the session key diversifier and the initialization vector from the IUT. Send an ENC_RSP packet in response.
 3. Lower Tester sends an LL_REJECT_EXT_IND PDU to the IUT with RejectOpcode set to "LL_ENC_REQ" and ErrorCode set to '0x06' indicating "PIN or Key missing".
 4. Upper Tester expects an HCI_Encryption_Change event from the IUT with the 'Status' parameter set to '0x06' indicating "PIN or key missing".
- Expected Outcome

Pass Verdict

Lower Tester receives an LL_ENC_REQ packet, containing the random number 'ediv', from the IUT.

Upper Tester receives an HCI_Encryption_Change event with the 'Status' parameter set to '0x06' indicating "PIN or key missing" from the IUT.

4.8.6.12 LL/SEC/MAS/BV-12-C [Master Start Encryption: Overlapping Procedure]

- Test Purpose

Tests that an IUT as master can complete the encryption start procedure correctly if an unexpected data channel PDU is received.

- Reference

[3] 5.1.3.1

- Initial Condition

Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and "ir").



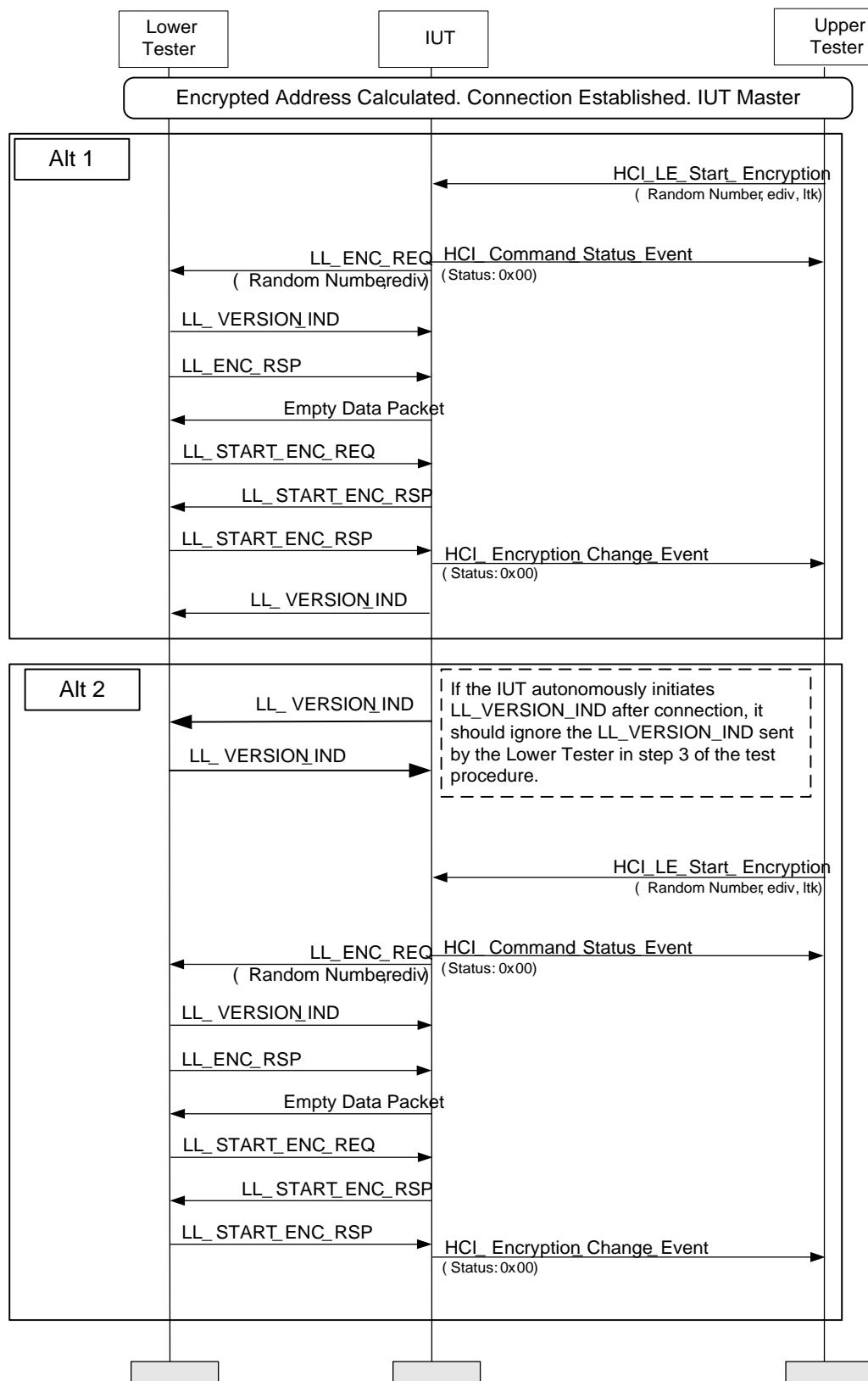


Figure 4.394: LL/SEC/MAS/BV-12-C [Master Start Encryption: Overlapping Procedure]



1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and ‘ediv’ from the preamble steps’ execution. Expect an HCI_Command_Status event in response.
2. The IUT sends an ENC_REQ packet including the parameters from step 1.
3. The Lower Tester sends a LL_VERSION_IND packet to the IUT followed by the LL_ENC_RSP.

Alternative 1:

1. The IUT should queue the response for the LL_VERSION_IND
2. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.
3. IUT sends the queued response to LL_VERSION_IND

Alternative 2:

1. The IUT should ignore the LL_VERSION_IND as the version information has already been exchanged prior to the start of the test procedure.
2. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.

- Expected Outcome

Pass verdict:

Alternative 1:

The IUT sends a queued response to the LL_VERSION_IND packet after the encryption start procedure has completed.

Alternative 2:

The IUT ignores the LL_VERSION_IND from step 3 as the version information has already been exchanged.

Alternatives 1 and 2:

The Upper Tester receives the Encryption_Change_Event from the IUT

4.8.6.13 LL/SEC/MAS/BV-13-C [Master Start Encryption: Overlapping Procedure with LL_SLAVE_FEATURE_REQ]

- Test Purpose

Tests that an IUT as master can complete the encryption start procedure correctly if an unexpected data channel PDU is received.

- Reference

[3] 5.1.3.1

- Initial Condition

Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

IUT must be capable of processing LL_SLAVE_FEATURE_REQ command

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables ‘er’ and “ir”).



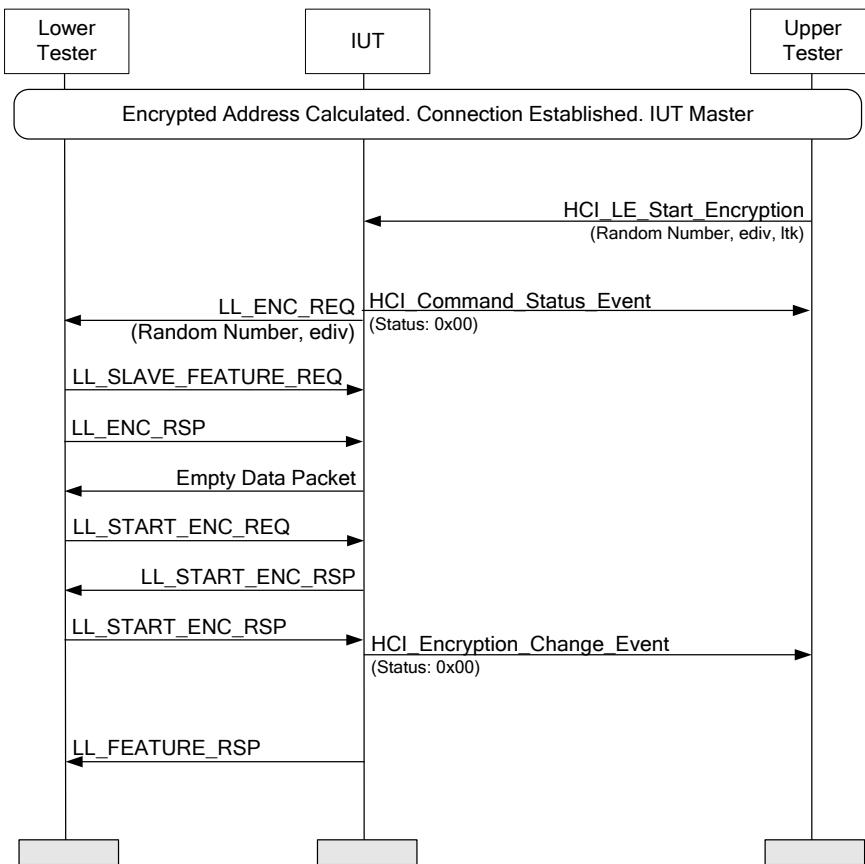


Figure 4.395: LL/SEC/MAS/BV-13-C [Master Start Encryption: Overlapping Procedure]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.
 2. The IUT sends an ENC_REQ packet including the parameters from step 1.
 3. The Lower Tester sends a LL_SLAVE_FEATURE_REQ packet to the IUT followed by the LL_ENC_RSP.
 4. The IUT should queue the response for the LL_SLAVE_FEATURE_REQ packet
 5. The encryption start procedure completes normally and the Upper Tester receives the HCI_Encryption_Change_Event.
 6. IUT sends the queued response to LL_SLAVE_FEATURE_REQ
- Expected Outcome

Pass verdict:

The IUT sends a queued LL_FEATURE_RSP packet after the encryption start procedure has completed.

The Upper Tester receives the Encryption_Change_Event from the IUT

4.8.6.14 LL/SEC/MAS/BV-14-C [Master Receiving unexpected PDU during encryption start]

- Test Purpose

Test that a Master IUT which has started the encryption procedure does not respond to an LL_VERSION_IND but instead drops the link.

The Lower Tester acts as a Slave.



- Reference

[3] 5.1.3.1

- Initial Condition

State: Connected Master (any advertising interval, any supported type of address, any advertising channel map, common connection interval, common timeout, any SCA value).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and "ir").

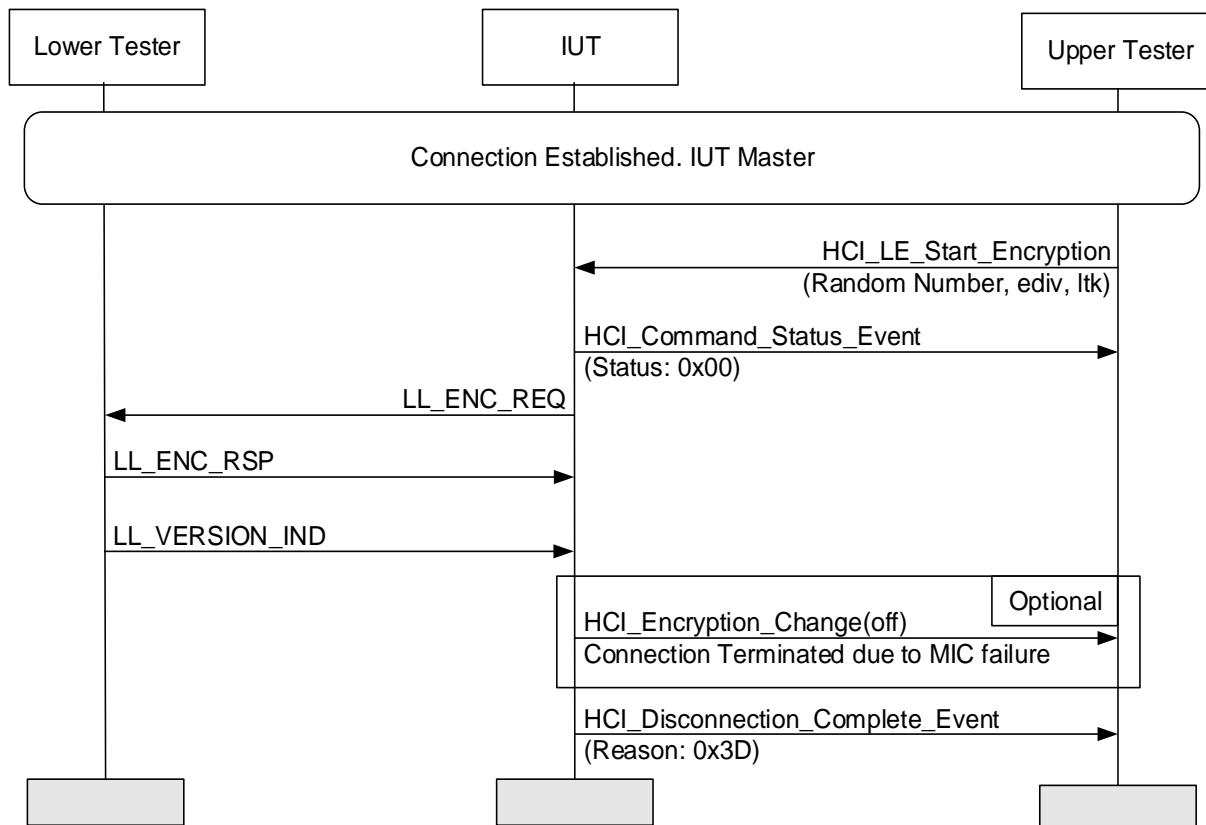


Figure 4.396: LL/SEC/MAS/BV-14-C [Master Receiving unexpected Data Channel PDU during encryption start]

1. In the preamble steps (connection establishment) use public address type for the Upper Tester and a supported type of address for the Lower Tester.
2. Upper Tester calculates the master portions of the session key diversifier, the initialization vector and a random number.
3. Upper Tester sends an HCI_Start_Encryption command, containing the random number, 'ediv' of master portions of the session key diversifier and the initialization vector to the Lower Tester. Expect an HCI_Command_Status_Event in response.
4. The Lower Tester sends LL_ENC_RSP packet, with the slave portions of the session key diversifier and the initialization vector.
5. The ENC_RSP packet is immediately followed by a LL_VERSION_IND packet from the Lower Tester. (In the same connection event if possible).
6. The IUT shall terminate the connection on receiving the LL_VERSION_IND packet.

7. Upper Tester receives an HCI_Disconnection_Complete event from the IUT, indicating termination from "Connection Terminated Due to MIC Failure (0x3D)."

- Expected Outcome

Pass verdict:

The IUT terminates the connection upon receiving the unexpected LL_VERSION_IND packet.

4.8.6.15 LL/SEC/MAS/BI-01-C [Master Encryption Setup: Missing Response]

- Test Purpose

Test that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send an encryption response packet.

The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'er' and 'ir').

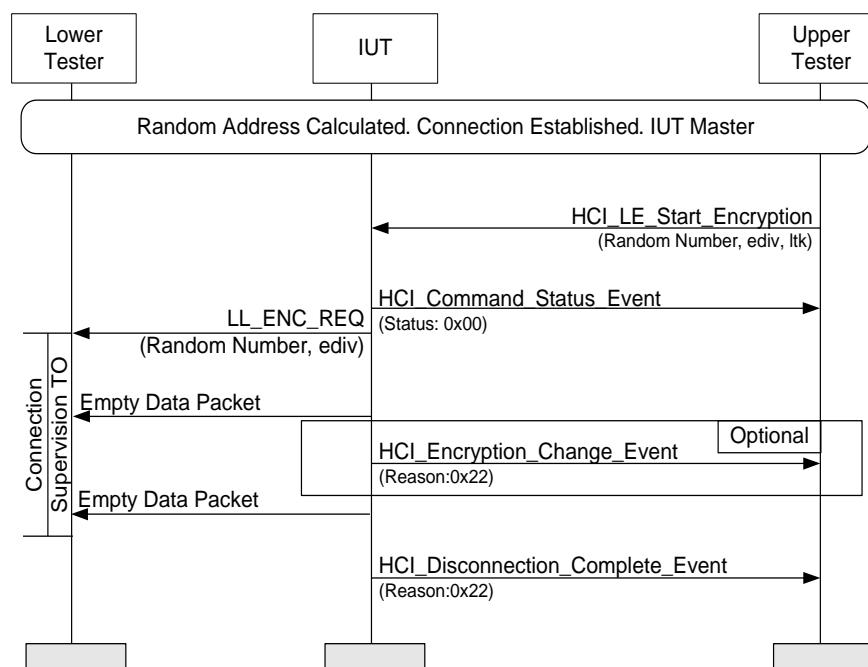


Figure 4.397: LL/SEC/MAS/BI-01-C [Master Encryption Setup: Missing Response]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.



2. Lower Tester expects an ENC_REQ packet including the parameters from step 1.
 3. Lower Tester acknowledges the encryption request, but does not respond with an ENC_RSP packet.
 4. The IUT sends empty data packets until the connection control timer expires.
 5. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping to maintain the connection.

4.8.6.16 LL/SEC/MAS/BI-03-C [Master Encryption Setup: Missing Request]

- Test Purpose

Test that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send a start encryption request packet.

The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout).

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'ltk' and "ir").

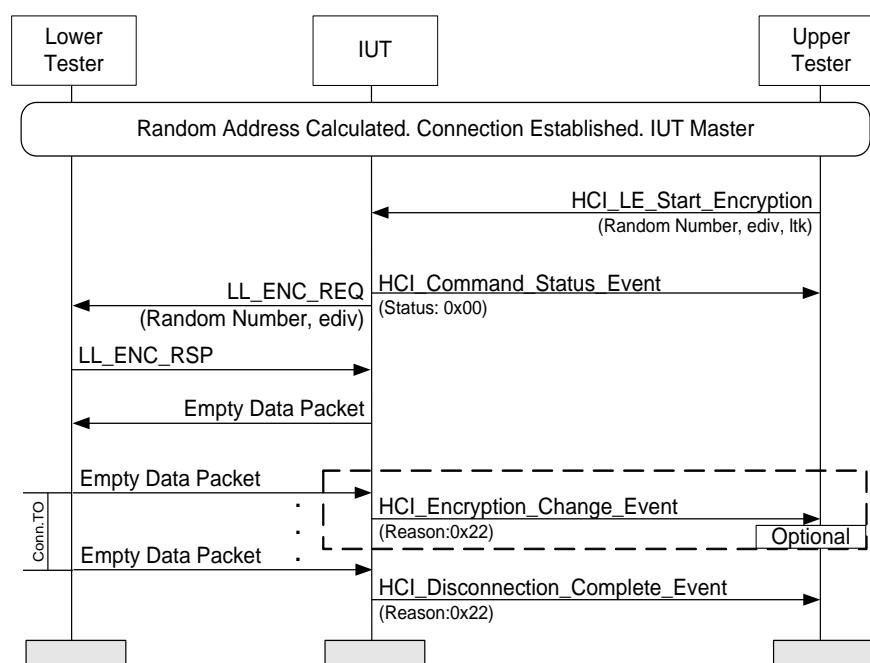


Figure 4.398: LL/SEC/MAS/BI-03-C [Master Encryption Setup: Missing Request]



1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.
2. Lower Tester expects an ENC_REQ packet including the parameters from step 1: with the master portions of the session key diversifier and the initialization vector.
3. Lower Tester acknowledges the encryption request and responds with an ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.
4. Lower Tester does not send START_ENC_REQ packet, but respond to transmissions with slave transmissions using the acknowledgement scheme.
5. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).

- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping to maintain the connection.

4.8.6.17 LL/SEC/MAS/BI-04-C [Master Encryption Setup: Missing Acknowledgement]

- Test Purpose

Tests that a master IUT is able to request encryption mode change from a slave Controller and recover from a slave device failing to send the second start encryption response packet.

The Lower Tester acts in the slave role, accepts the encryption mode setup request from the IUT, and then omits packets from the setup sequence.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure with the common connection parameters and using the common variables for the encryption keys (variables 'ltk' and 'ir').



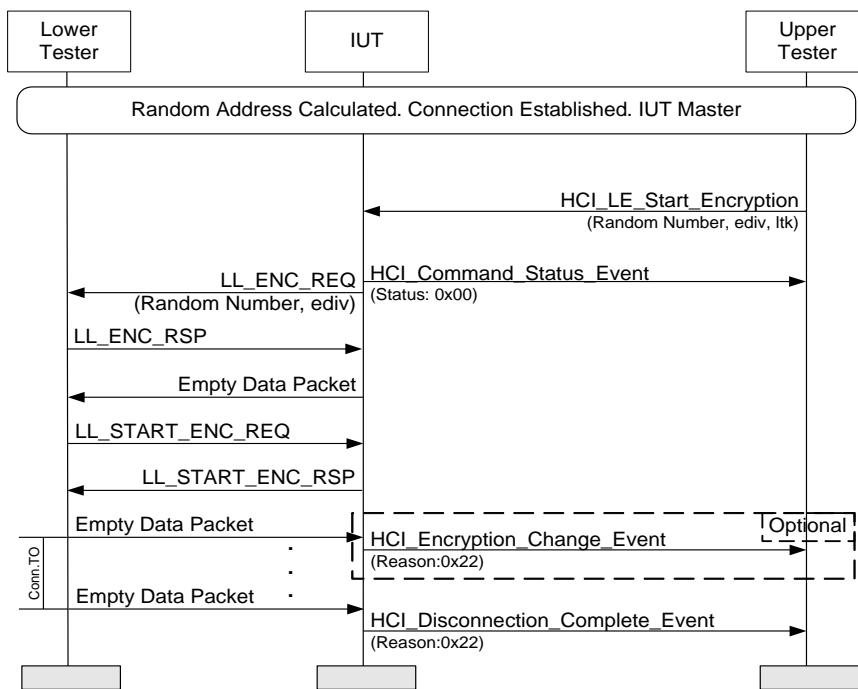


Figure 4.399: LL/SEC/MAS/BI-04-C [Master Encryption Setup: Missing Acknowledgement]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.
 2. Lower Tester expects an ENC_REQ packet including the parameters from step 1: with the master portions of the session key diversifier and the initialization vector. Lower Tester acknowledges the encryption request and responds with an ENC_RSP packet with the slave portions of the session key diversifier and the initialization vector.
 3. Lower Tester calculates the session key diversifier and initialization vector. Calculates the diversifier, using the parameters received in step 2. Calculates the long term key from the diversifier using the diversifier hiding key.
 4. Lower Tester expects an empty DATA packet from the IUT and responds with an START_ENC_REQ packet.
 5. Lower Tester expects an START_ENC_RSP packet encrypted from the IUT. Once the packet is received acknowledges the packet but does not respond with an START_ENC_RSP packet.
 6. Lower Tester sends empty data packet. The IUT responds to transmissions with slave transmissions using the acknowledgement scheme until the connection control timer expires.
 7. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, indicating termination due to LL Response Timeout (0x22).
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT stopping to maintain the connection.

4.8.6.18 LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC]

- Test Purpose

Tests that a master IUT terminates a connection upon a MIC failure in a packet received from a slave device.



The Lower Tester acts in the slave role in an encrypted connection, first transferring data with valid packets, then corrupts the packet contents to cause termination.

- Reference
- [3] 3.1, 5.1.3
- Initial Condition

State: Encrypted Master Connection (common identity root, common encryption root, any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.

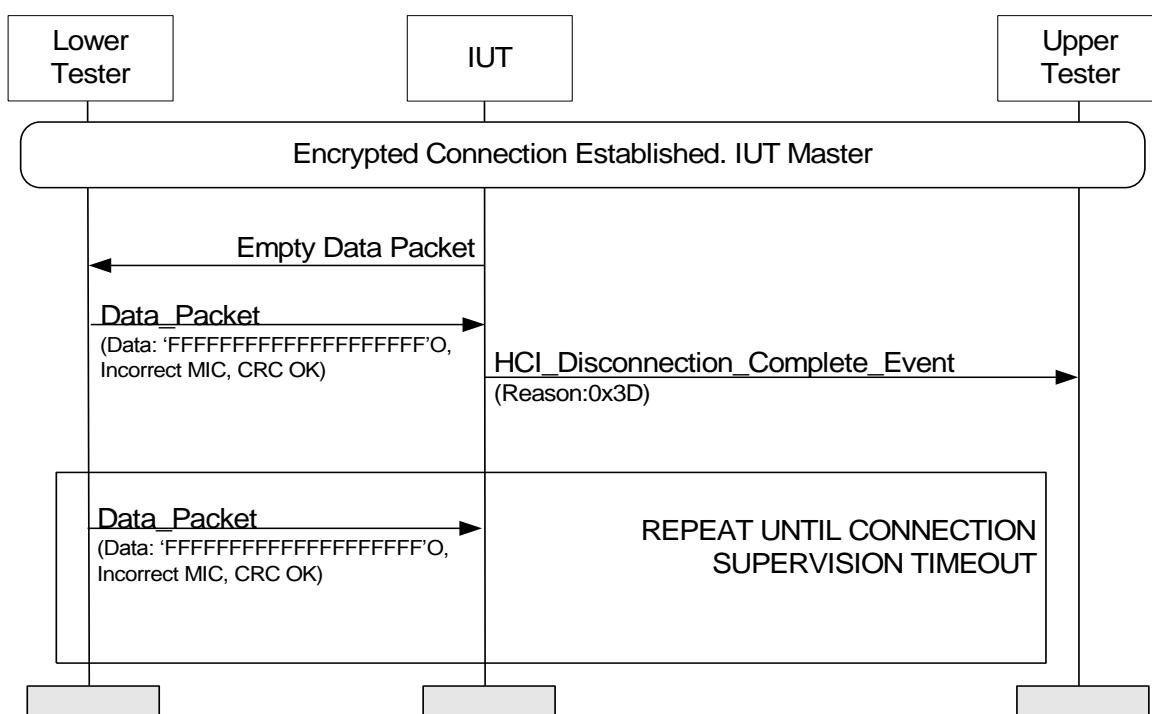


Figure 4.400: LL/SEC/MAS/BI-05-C [Master MIC Failure: Corrupt MIC]

1. Lower Tester expects a DATA packet from the IUT and sends a DATA packet in response with 10 bytes of data, flipping a bit in the MIC calculated from the unencrypted packet.
 2. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
 3. Lower Tester continues slave transmissions.
 4. Repeat Step 3 until the IUT stops master transmissions up to a time equal to the connection supervision timeout value.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT terminating the connection.

The IUT reports the termination with an HCI event.



4.8.6.19 LL/SEC/MAS/BI-06-C [Master MIC Failure: Corrupt Header]

- Test Purpose

Test that a master IUT terminates a connection upon a MIC failure in a packet received from a slave device.

The Lower Tester acts in the slave role in an encrypted connection, first transferring data with valid packets, then corrupts the packet contents to cause termination.

- Reference

[3] 3.1, 5.1.3

- Initial Condition

State: Encrypted Master Connection (common identity root, common encryption root, any scan interval, any scan window, any type of peer address, Lower Tester address, public own address, connection interval, common slave latency, common timeout)

- Test Procedure

Execute the test procedure starting from an encrypted connection setup using the common connection parameters and encryption key variables.

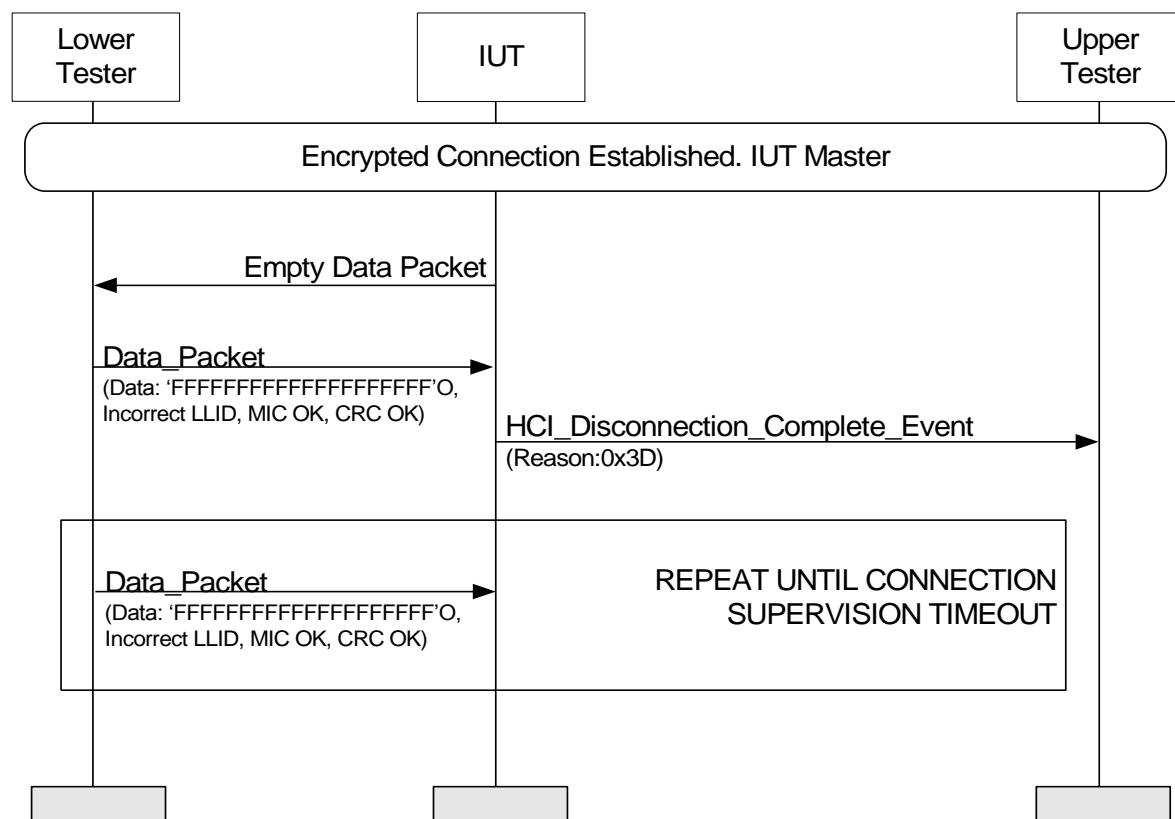


Figure 4.401: LL/SEC/MAS/BI-06-C [Master MIC Failure: Corrupt Header]

- Lower Tester expects a DATA packet from the IUT and transmits a DATA packet in response with 10 bytes of data, flipping a single bit in the header (LLID or RFU) after MIC calculation from the unencrypted packet. Upper Tester expects an HCI_Disconnection_Complete event from the IUT, with the reason code indicating MIC failure.
- Lower Tester continues slave transmissions.



3. Repeat step 2 until the IUT stops master transmissions up to a time equal to the connection supervision timeout value.
- Expected Outcome

Pass Verdict

The test procedure executes successfully, with the IUT terminating the connection,

The IUT reports the termination with an HCI event.

4.8.6.20 LL/SEC/MAS/BI-07-C [Master Pause Encryption Sending Data]

- Test Purpose

Test that a master IUT terminates the established connection when a data packet is received during the encryption pause procedure.

The Lower Tester acts as in the slave role in an encrypted connection, first accepting the encryption pause request from the IUT, then sends a data packet during the encryption pause procedure to cause termination.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, selected type of own address, connection interval, common slave latency, common timeout).



- Test Procedure

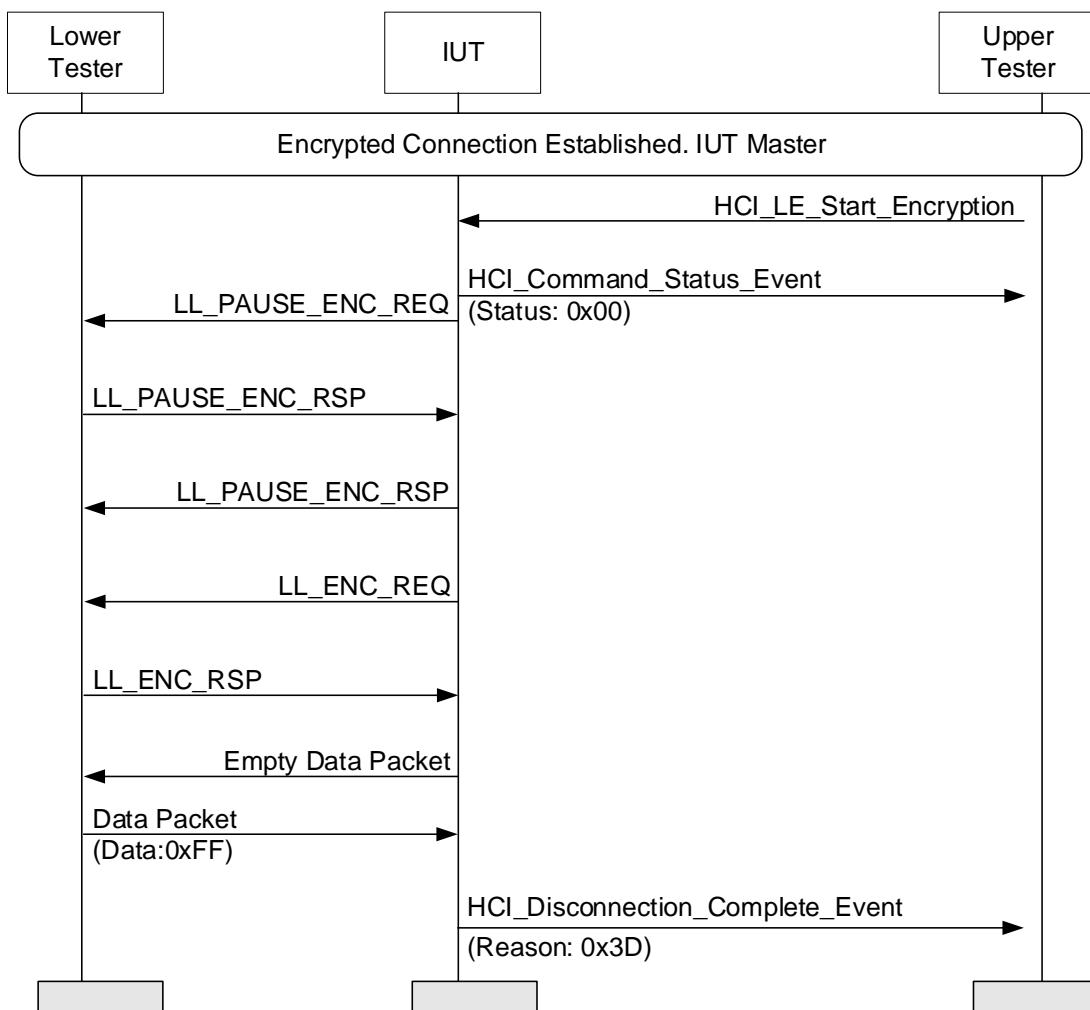


Figure 4.402: LL/SEC/MAS/BI-07-C [Master Pause Encryption Sending Data]

1. As preamble: an encrypted connection is established (IUT acts as master)
2. Upper Tester sends an **HCI_LE_Start_Encryption** command and expects an **HCI_Command_Status** in response.
3. Lower Tester expects an encrypted **PAUSE_ENC_REQ** packet and sends an encrypted **PAUSE_ENC_RSP** packet in response.
4. Lower Tester expects an unencrypted **PAUSE_ENC_RSP** packet from the IUT. At this point the connection stops being encrypted.
5. Lower Tester expects a **LL_ENC_REQ** packet to re-enable encryption using a new session key.
6. Lower Tester sends an **LL_ENC_RSP** packet.
7. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
8. Upper Tester expects an **HCI_Disconnection_Complete** event with status set to “Connection Terminated Due to MIC Failure”.

An IUT that conforms to Core Specification v4.0 may send a **TERMINATE_IND** PDU with error code “Connection Terminated Due to MIC Failure (0x3D)” to the Lower Tester between steps 7 and 8.

- Expected Outcome

Pass Verdict

The IUT sends an HCI_Disconnection_Complete event with status set to “Connection Terminated Due to MIC Failure”.

4.8.6.21 LL/SEC/MAS/BI-08-C [Master Encryption: Sending Data and Not Response]

- Test Purpose

Test that a master IUT terminates the established connection when the procedure response timeout timer expires during encryption setup process.

The Lower Tester acts as in the slave role, first accepting the encryption mode setup request from the IUT, then sends a data packet during the encryption start procedure and then never sending LL_ENC_RSP packet, thus triggering connection termination due to procedure response timeout timer expiring on IUT.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).

- Test Procedure

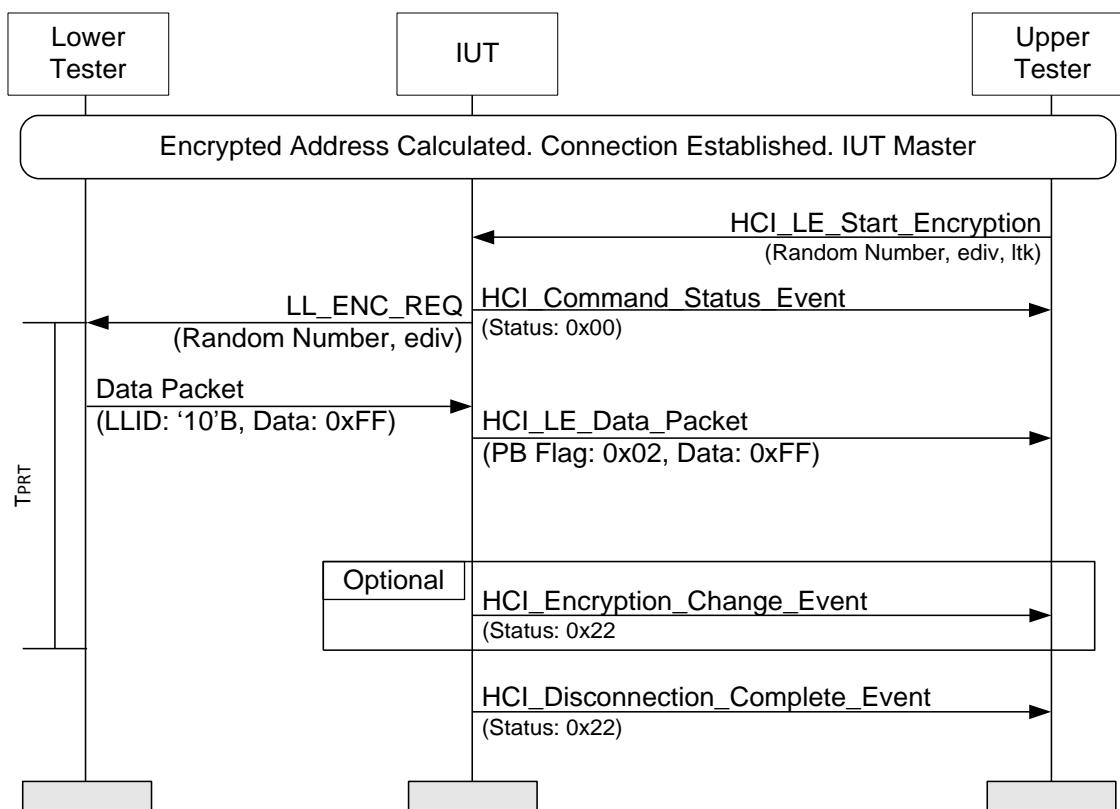


Figure 4.403: LL/SEC/MAS/BI-08-C [Master Encryption: Sending Data and Not Response]

1. Upper Tester sends an HCI_LE_Start_Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCI_Command_Status event in response.
 2. Lower Tester expects an ENC_REQ packet including the parameters from step 1.
 3. Lower Tester acknowledges the encryption request, but does not respond with an ENC_RSP packet.
 4. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
 5. Upper Tester expects an HCI_Disconnection_Complete event with status set to "LL response timeout".
- Expected Outcome

Pass Verdict

The IUT sends an HCI_Disconnection_Complete event with status set to "LL response timeout".

4.8.6.22 LL/SEC/MAS/BI-09-C [Master Encryption: Sending Data and Not Request]

- Test Purpose

Test that a master IUT terminates the established connection when a data packet is received during the encryption start procedure instead of a start encryption request packet.

The Lower Tester acts as in the slave role, first accepting the encryption mode setup request from the IUT, then sends a data packet during the setup sequence.

- Reference

[3] 5.1.3

- Initial Condition

State: Encryption Keys Calculated (common identity root, common encryption root) AND Connected Master (any scan interval, any scan window, selected type of peer address, Lower Tester address, public own address, common connection interval, common slave latency, common timeout).



- Test Procedure

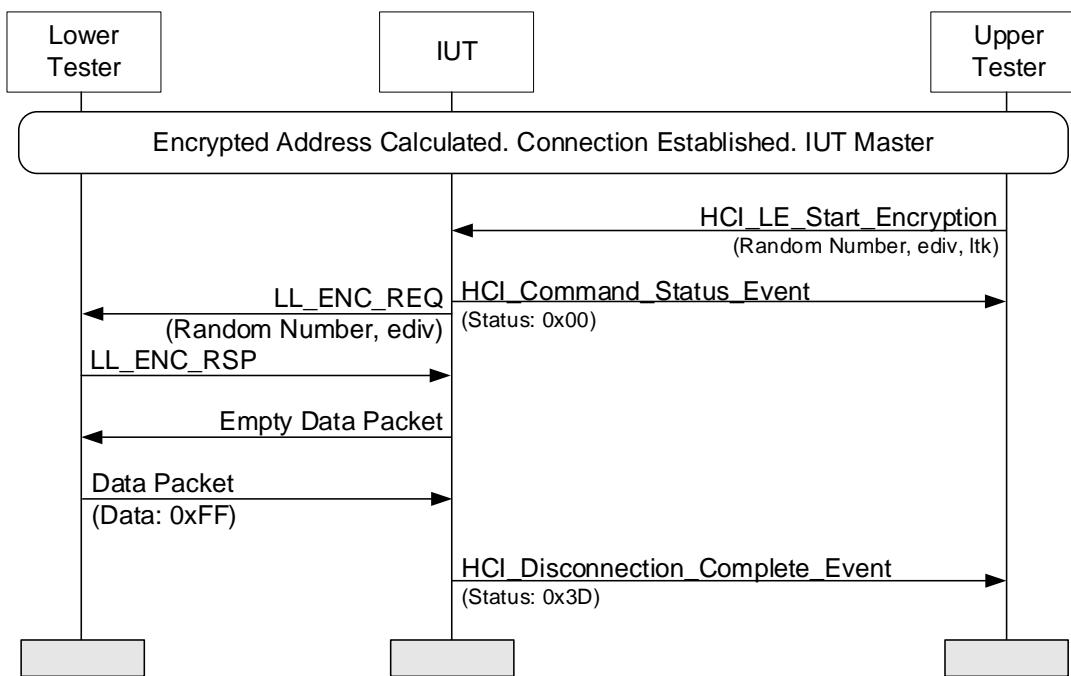


Figure 4.404: LL/SEC/MAS/BI-09-C [Master Encryption: Sending Data and Not Request]

1. Upper Tester sends an HCl LE Start Encryption command to the IUT including the connection handle, the random number and 'ediv' from the preamble steps' execution. Expect an HCl Command Status event in response.
2. Lower Tester expects an ENC_REQ packet including the parameters from step 1.
3. Lower Tester acknowledges the encryption request and responds with an ENC_RSP packet.
4. Configure Lower Tester to send a DATA packet to the IUT containing a byte as data (0xFF).
5. Upper Tester expects an HCl Disconnection Complete event with status set to "Connection Terminated Due to MIC Failure".

An IUT that conforms to Core Specification v4.0 may send a TERMINATE_IND PDU with error code "Connection Terminated Due to MIC Failure (0x3D)" to the Lower Tester between steps 4 and 5.

- Expected Outcome

Pass Verdict

The IUT sends an HCl Disconnection Complete event with status set to "Connection Terminated Due to MIC Failure".



4.9 Data Flow

4.9.1 Both Connected Roles

4.9.1.1 [Transmit Fragmented L2CAP Header]

Test that the IUT correctly transmits packets with fragmented L2CAP headers.

- Test Case IDs

[LL/DFL/SLA/BV-01-C](#)

[LL/DFL/MAS/BV-01-C](#)

- Reference

[\[11\] 5.4.2](#)

[\[12\] 7.2.1](#)

- Initial Condition

State: Connected in the relevant role (any scan interval, any scan window, public peer address, lower tester address, supported type of own address, connection interval, common slave latency, common timeout) to the Lower Tester.



- Test Procedure

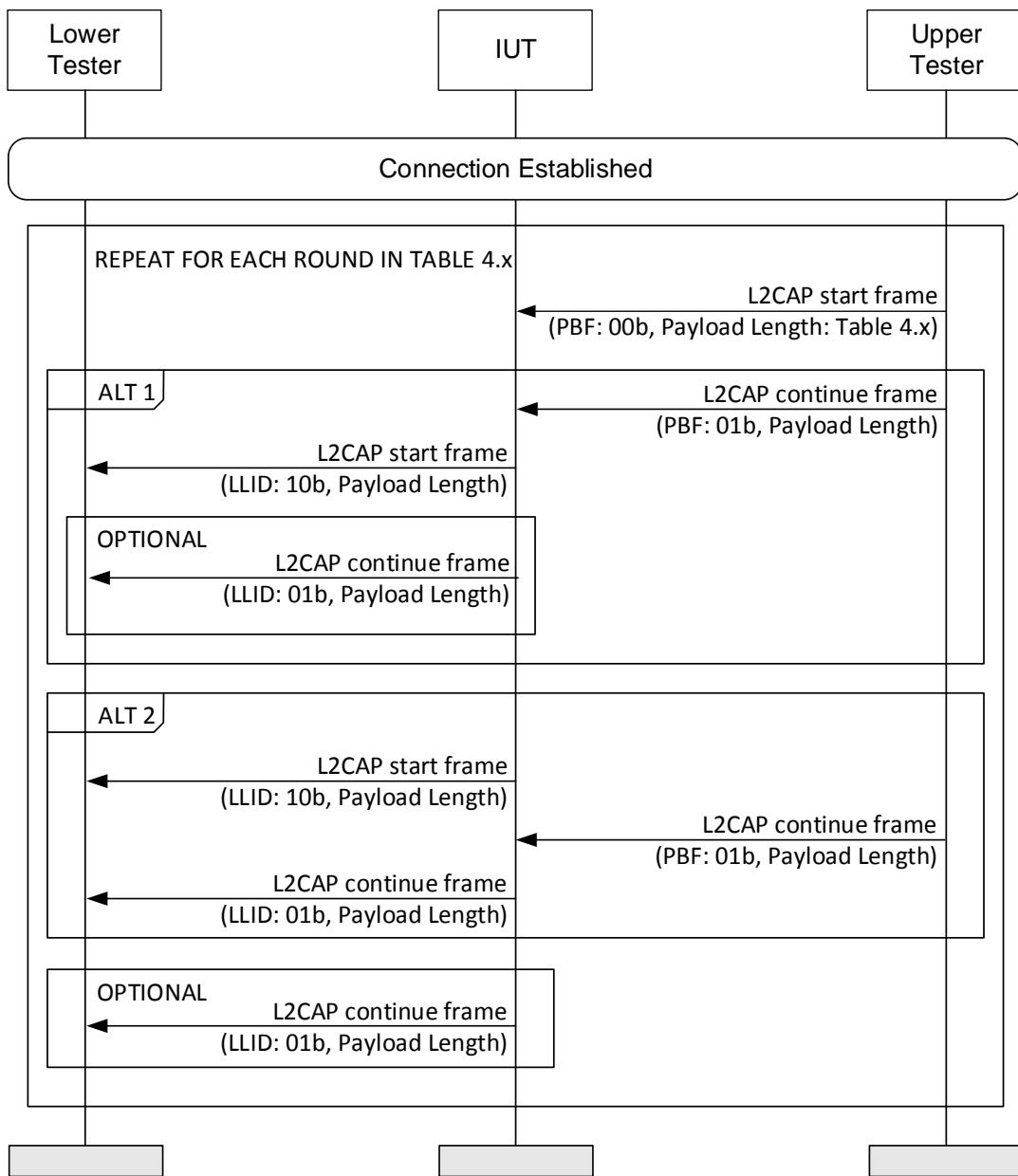


Figure 4.405: [Transmit Fragmented L2CAP Header]

1. The Upper Tester sends a L2CAP frame of 28 octets to the IUT with the start fragment containing a Payload length according to [Table 4.62](#) and the rest in a single continue fragment.

Round	Payload Length (octets) (Step b)
1	0
2	1
3	2
4	3



5	4
6	5

Table 4.62: Payload length for each round

2. The Lower Tester receives the L2CAP frame unaltered, possibly unfragmented or fragmented differently.
- Expected Outcome

Pass Verdict

The Lower Tester receives the L2CAP frame unaltered, irrespective of fragmentation.

4.9.1.2 [Receive Fragmented L2CAP Header]

Test that the IUT correctly receives packets with fragmented L2CAP headers.

- Test Case IDs

[LL/DFL/SLA/BV-02-C](#)

[LL/DFL/MAS/BV-02-C](#)

- Reference

[11] 5.4.2

[12] 7.2.1

- Initial Condition

State: Connected in the relevant role (any scan interval, any scan window, public peer address, lower tester address, supported type of own address, connection interval, common slave latency, common timeout) to the Lower Tester.



- Test Procedure

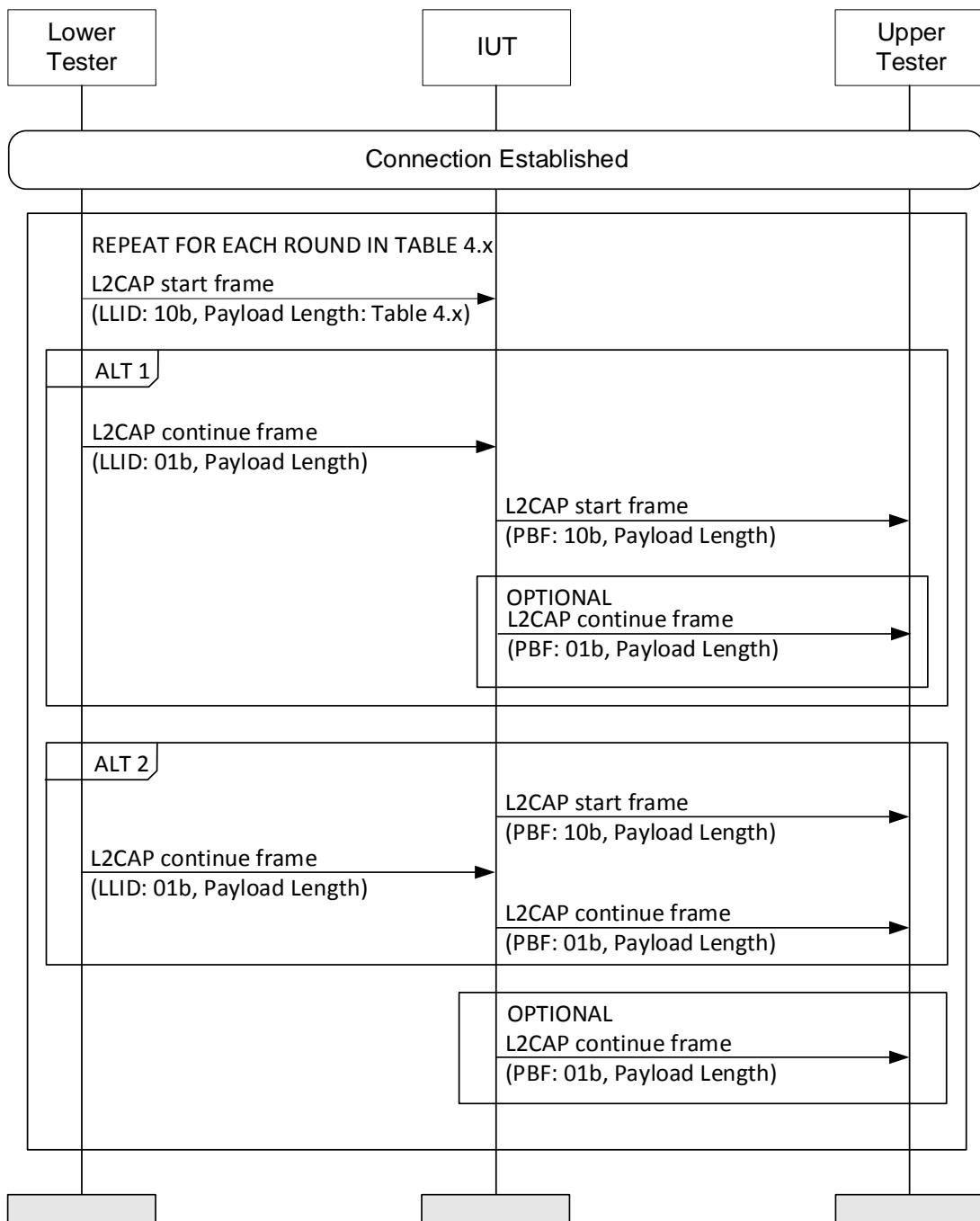


Figure 4.406: [Receive Fragmented L2CAP Header]

- The Lower Tester sends an L2CAP frame of 28 octets to the IUT with the start fragment containing a Payload length according to [Table 4.63](#) and the rest in a single continue fragment.

Round	Payload Length (octets) (Step b)
1	1
2	2

Round	Payload Length (octets) (Step b)
3	3
4	4
5	5

Table 4.63: Payload length for each round

2. The Upper Tester receives the L2CAP frame unaltered, possibly unfragmented or fragmented differently.
- Expected Outcome

Pass Verdict

The Upper Tester receives the L2CAP frame unaltered, irrespective of fragmentation.



5 Test Case Mapping

The Test Case Mapping Table (TCMT) maps test cases to specific capabilities in the ICS. Profiles, protocols and services may define multiple roles, and it is possible that a product may implement more than one role. The product shall be tested in all roles for which support is declared in the ICS document. For products which support more than one role, a separate TCMT shall be filled out for each role, and separate tests shall be conducted for each role.

The columns for the TCMT are defined as follows:

Item: contains a y/x reference, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS Proforma for LL in [4]. If the item is defined with Protocol, Profile or Service abbreviation before y/x, the table and feature number referenced are defined in the abbreviated ICS proforma document.

Feature: recommended to be the primary feature defined in the ICS being tested or may be the test case name.

Test Case(s): the applicable test case identifiers required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported.

For purpose and structure of the ICS/IXIT proforma and instructions for completing the ICS/IXIT proforma refer to the Bluetooth ICS and IXIT proforma document.

Item	Feature	Test Case(s)
LL 3/1	Non-Connectable Advertising Events	LL/DDI/ADV/BV-01-C
LL 3/1 AND LL 3/3	Advertising Data: Non-Connectable	LL/DDI/ADV/BV-03-C
LL 3/2	Undirected Advertising Events	LL/DDI/ADV/BV-02-C
LL 3/2 AND LL 3/3	Advertising Data: Undirected	LL/DDI/ADV/BV-04-C
LL 3/2 AND LL 3/6	Scan Request: Undirected Connectable Scan Request Invalid Address	LL/DDI/ADV/BV-05-C LL/ENC/ADV/BI-01-C
LL 3/2 AND LL 3/6 AND LL 3/8	Scan Request Device Filtering	LL/DDI/ADV/BV-08-C
LL 3/2 AND LL 3/6 AND LL 3/8 AND LL 6/1	Advertising With Static Address	LL/SEC/ADV/BV-01-C
LL 3/1 AND LL 2/4	Privacy – Non-connectable Undirected Advertising with private address	LL/SEC/ADV/BV-02-C LL/SEC/ADV/BV-03-C
LL 3/5 AND LL 2/4	Privacy - Scannable Advertising with private address	LL/SEC/ADV/BV-04-C LL/SEC/ADV/BV-05-C
LL 3/5 AND LL 2/6	Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Scannable Advertising	LL/SEC/ADV/BV-15-C
LL 3/5 AND LL 2/6 AND LL 2/5	Network Privacy - Scannable Advertising, resolvable private address, Ignore scanner RPA	LL/SEC/ADV/BV-21-C
LL 3/5 AND LL 2/7	Device Privacy Mode, Scannable Advertising	LL/SEC/ADV/BV-18-C



Item	Feature	Test Case(s)
LL 3/2 AND LL 2/4	Privacy - Undirected Connectable Advertising	LL/SEC/ADV/BV-06-C LL/SEC/ADV/BV-07-C LL/SEC/ADV/BV-08-C LL/SEC/ADV/BV-09-C LL/SEC/ADV/BV-10-C
LL 3/2 AND LL 2/6	Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Undirected Connectable Advertising	LL/SEC/ADV/BV-16-C
LL 3/2 AND LL 2/7	Device Privacy Mode, Undirected Connectable Advertising	LL/SEC/ADV/BV-19-C
LL 3/4 AND LL 2/4	Privacy - Directed Connectable Advertising	LL/SEC/ADV/BV-11-C LL/SEC/ADV/BV-12-C LL/SEC/ADV/BV-13-C LL/SEC/ADV/BV-14-C
LL 3/4 AND LL 2/6	Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Directed Connectable Advertising	LL/SEC/ADV/BV-17-C
LL 3/4 AND LL 2/7	Device Privacy Mode, Directed Connectable Advertising	LL/SEC/ADV/BV-20-C
LL 3/2 AND LL 3/7	Connection Request Invalid CRC Connection Request	LL/DDI/ADV/BI-02-C LL/DDI/ADV/BV-06-C
LL 3/2 AND LL 3/7 AND LL 3/6	Scan Request Connection Request	LL/DDI/ADV/BV-07-C
LL 3/2 AND LL 3/7 AND LL 1/4	Advertiser IUT, Slave role	LL/CON/ADV/BI-01-C LL/ENC/ADV/BI-02-C LL/CON/ADV/BV-03-C LL/CON/ADV/BV-01-C
LL 3/2 AND LL 3/7 AND LL 6/1 AND LL 6/2 AND LL 8/3	Accepting Connections With Hop Lengths	LL/FRH/ADV/BV-01-C
LL 3/2 AND LL 3/7 AND LL 3/4	Directed Advertising Events	LL/DDI/ADV/BV-11-C
LL 3/2 AND LL 3/7 AND LL 3/8	Connection Request Device Filtering	LL/DDI/ADV/BV-09-C
LL 3/2 AND LL 3/7 AND LL 6/16	Accepting Connections Timeout	LL/CON/ADV/BV-02-C
LL 3/4 AND LL 3/7 AND LL 6/1 AND LL 6/2	Directed Advertising Connection	LL/CON/ADV/BV-04-C
LL 3/5	Scannable Advertising Events	LL/DDI/ADV/BV-15-C
LL 3/5 AND LL 3/3	Advertising Data: Scannable	LL/DDI/ADV/BV-16-C
LL 3/5 AND LL 3/6	Scan Request: Scannable	LL/DDI/ADV/BV-17-C
LL 3/5 AND LL 3/6 AND LL 3/8	Device Filtering: Scannable	LL/DDI/ADV/BV-18-C
LL 3/4a AND LL 3/7	Low Duty Cycle Directed Advertising Events	LL/DDI/ADV/BV-19-C
LL 3/6 AND (LL 3/2 OR LL 3/5)	Scan Request Invalid CRC	LL/DDI/ADV/BI-01-C



Item	Feature	Test Case(s)
LL 3/2 AND LL 3/7 AND LL 1/4 AND LL 9/10	Accepting Connection Requests, Channel Selection Algorithm #2	LL/CON/ADV/BV-07-C LL/CON/ADV/BV-08-C LL/CON/ADV/BV-09-C LL/CON/ADV/BV-10-C
LL 3/7 AND LL 8/3 AND NOT LL 9/10	Accepting Connection Requests, Support Data channel selection algorithm	LL/CON/ADV/BV-11-C
LL 3/9	Extended Advertising	LL/DDI/ADV/BV-47-C LL/DDI/ADV/BV-27-C LL/DDI/ADV/BV-28-C
LL 3/9 AND LL 9/9	Extended Advertising – LE Coded PHY	LL/DDI/ADV/BV-48-C
LL 3/9 AND LL 9/7	Extended Advertising – LE 2M PHY	LL/DDI/ADV/BV-49-C
LL 3/9 AND LL 3/12	Extended Advertising, Sending Tx Power in Advertisements	LL/DDI/ADV/BV-34-C
LL 3/9 AND LL 3/5	Extended Advertising, Scannable	LL/TIM/ADV/BV-03-C LL/TIM/ADV/BV-04-C
LL 3/9 AND LL 3/5 AND SUM ICS 21/16	Extended Advertising, Scannable, without ADI	LL/DDI/ADV/BV-25-C
LL 3/9 AND LL 3/5 AND NOT SUM ICS 21/16	Extended Advertising, Scannable, with ADI	LL/DDI/ADV/BV-45-C
LL 3/9 AND LL 3/3	Extended Advertising, Legacy PDUs with Data	LL/DDI/ADV/BV-21-C LL/DDI/ADV/BV-22-C
LL 3/9 AND LL 3/7	Extended Advertising, Connectable	LL/CON/ADV/BV-05-C
LL 3/9 AND LL 3/7	Extended Advertising, Legacy PDUs, Connectable	LL/CON/ADV/BV-06-C
LL 3/10	Periodic Advertising	LL/DDI/ADV/BV-26-C
LL 3/10 AND LL 3/11	Periodic Advertising, Multiple Sets	LL/DDI/ADV/BV-33-C
LL 3/11	Extended Advertising, Multiple Sets, LE 1M PHY	LL/DDI/ADV/BV-29-C LL/DDI/ADV/BV-32-C LL/DDI/ADV/BV-35-C
LL 3/11 AND LL 9/9	Extended Advertising, Multiple Sets, LE Coded PHY	LL/DDI/ADV/BV-30-C LL/DDI/ADV/BV-31-C
LL 2/1 AND LL 3/13 AND LL 9/18	Connectionless CTE Transmitter, Public Device Addresses, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD)	LL/DDI/ADV/BV-36-C
LL 2/1 AND LL 3/13 AND LL 9/19	Connectionless CTE Transmitter, Public Device Addresses, No Antenna Switching During Constant Tone Extension Transmission (AoA)	LL/DDI/ADV/BV-37-C
LL 2/1 AND LL 3/13	Connectionless CTE Transmitter, Public Device Addresses,	LL/DDI/ADV/BV-39-C
LL 3/10 AND NOT SUM ICS 21/16	Periodic Advertising, SyncInfo Validation	LL/DDI/ADV/BV-43-C
LL 11/2	Connectionless CTE Receiver, Public Addresses	LL/DDI/SCN/BV-29-C LL/DDI/SCN/BV-31-C



Item	Feature	Test Case(s)
LL 11/2 AND LL 9/21	Connectionless CTE Receiver, Public Addresses, 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)	LL/DDI/SCN/BV-30-C LL/DDI/SCN/BV-32-C
LL 11/2	Connectionless CTE Receiver, Public Addresses	LL/DDI/SCN/BV-35-C LL/DDI/SCN/BV-36-C
LL 4/1	Passive Scanning Invalid Address	LL/ENC/SCN/BI-01-C
LL 4/1 AND LL 4/2	Passive Scanning Invalid CRC Passive Scanning: Non-connectable Passive Scanning: Undirected Events Passive Scanning: Directed Events Passive Scanning: Discoverable Events	LL/DDI/SCN/BI-02-C LL/DDI/SCN/BV-01-C LL/DDI/SCN/BV-10-C LL/DDI/SCN/BV-11-C LL/DDI/SCN/BV-12-C
LL 4/1 AND LL 4/2 AND LL 4/5	Passive Scanning Device Filtering	LL/DDI/SCN/BV-02-C
LL 4/2 AND LL 4/3	Active Scanning Invalid CRC Active Scanning	LL/DDI/SCN/BI-01-C LL/DDI/SCN/BV-03-C
LL 2/2 AND LL 4/2 AND LL 4/3	Random Address Scanning	LL/SEC/SCN/BV-01-C
LL 4/2 AND LL 4/3 AND LL 4/4	Scanning For Advertiser Types	LL/DDI/SCN/BV-05-C
LL 4/3	Active Scanning Invalid Address	LL/ENC/SCN/BI-02-C
LL 4/3 AND LL 4/2 AND LL 4/5	Active Scanning Device Filtering	LL/DDI/SCN/BV-04-C
LL 4/7	Extended Scanning	LL/DDI/SCN/BV-19-C LL/DDI/SCN/BV-23-C
LL 4/7 AND LL 4/3	Extended Scanning, Active	LL/DDI/SCN/BV-20-C LL/DDI/SCN/BV-24-C LL/TIM/SCN/BV-01-C
LL 4/7 AND LL 9/9 AND LL 4/3	Extended Scanning, Active, LE Coded PHY	LL/TIM/SCN/BV-02-C LL/TIM/SCN/BV-03-C
LL 4/7 AND LL 4/3 AND LL 4/5 AND LL 2/4	Extended Scanning, Active, Filtering Policies, Generation of private addresses	LL/DDI/SCN/BV-33-C
LL 4/8 OR LL 11/1	Scanning for Periodic Advertising or Synchronizing to Periodic Advertising	LL/DDI/SCN/BV-21-C LL/DDI/SCN/BV-25-C LL/DDI/SCN/BV-37-C
LL 11/1 AND HCI 6/37	Synchronizing to Periodic Advertising LE Set Periodic Advertising Receive Enable Command	LL/DDI/SCN/BV-38-C
LL 6/26	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Slave Role, LE 1M	LL/CON/SLA/BV-88-C
LL 6/26 AND LL 9/7 AND NOT LL 9/9	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Slave Role, LE 2M	LL/CON/SLA/BV-90-C LL/CON/SLA/BV-92-C
LL 6/26 AND LL 9/9	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Slave Role, LE Coded	LL/CON/SLA/BV-89-C LL/CON/SLA/BV-91-C



Item	Feature	Test Case(s)
LL 6/27	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Slave Role, LE 1M	LL/CON/SLA/BV-93-C
LL 6/27 AND LL 9/7 AND NOT LL 9/9	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Slave Role, LE 2M	LL/CON/SLA/BV-95-C LL/CON/SLA/BV-97-C
LL 6/27 AND LL 9/9	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Slave Role, LE Coded	LL/CON/SLA/BV-94-C LL/CON/SLA/BV-96-C
LL 6/28	Accepting Periodic Advertising Sync Transfer, Slave Role, LE 1M	LL/CON/SLA/BV-98-C LL/CON/SLA/BV-103-C LL/CON/SLA/BV-104-C LL/CON/SLA/BV-105-C LL/CON/SLA/BV-106-C LL/CON/SLA/BV-107-C LL/CON/SLA/BV-108-C
LL 6/28 AND LL 9/7 AND NOT LL 9/9	Accepting Periodic Advertising Sync Transfer, Slave Role, LE 2M	LL/CON/SLA/BV-100-C LL/CON/SLA/BV-102-C
LL 6/28 AND LL 9/9	Accepting Periodic Advertising Sync Transfer, Slave Role, LE Coded	LL/CON/SLA/BV-99-C LL/CON/SLA/BV-101-C
LL 6/29	Sending Long Control PDUs, Slave Role, LE 1M	LL/CON/SLA/BV-112-C LL/CON/SLA/BV-115-C LL/CON/SLA/BV-118-C
LL 6/29 AND LL 9/7 AND NOT LL 9/9	Sending Long Control PDUs, Slave Role, LE 2M	LL/CON/SLA/BV-113-C LL/CON/SLA/BV-116-C LL/CON/SLA/BV-119-C
LL 6/29 AND LL 9/9	Sending Long Control PDUs, Slave Role, LE Coded	LL/CON/SLA/BV-114-C LL/CON/SLA/BV-117-C LL/CON/SLA/BV-120-C
LL 6/30	Receiving Long Control PDUs, Slave Role, LE 1M	LL/CON/SLA/BV-109-C
LL 6/30 AND LL 9/7 AND NOT LL 9/9	Receiving Long Control PDUs, Slave Role, LE 2M	LL/CON/SLA/BV-110-C
LL 6/30 AND LL 9/9	Receiving Long Control PDUs, Slave Role, LE Coded	LL/CON/SLA/BV-111-C
LL 7/26	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Master Role, LE 1M	LL/CON/MAS/BV-84-C
LL 7/26 AND LL 9/7 AND NOT LL 9/9	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Master Role, LE 2M	LL/CON/MAS/BV-86-C LL/CON/MAS/BV-88-C
LL 7/26 AND LL 9/9	Initiating Periodic Advertising Sync Transfer for Local Periodic Advertising, Master Role, LE Coded	LL/CON/MAS/BV-85-C LL/CON/MAS/BV-87-C
LL 7/27	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Master Role, LE 1M	LL/CON/MAS/BV-89-C



Item	Feature	Test Case(s)
LL 7/27 AND LL 9/7 AND NOT LL 9/9	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Master Role, LE 2M	LL/CON/MAS/BV-91-C LL/CON/MAS/BV-93-C
LL 7/27 AND LL 9/9	Initiating Periodic Advertising Sync Transfer Procedure for Remote Periodic Advertising, Master Role, LE Coded	LL/CON/MAS/BV-90-C LL/CON/MAS/BV-92-C
LL 7/28	Accepting Periodic Advertising Sync Transfer, Master Role, LE 1M	LL/CON/MAS/BV-94-C LL/CON/MAS/BV-99-C LL/CON/MAS/BV-100-C LL/CON/MAS/BV-101-C LL/CON/MAS/BV-102-C LL/CON/MAS/BV-103-C LL/CON/MAS/BV-104-C
LL 7/28 AND LL 9/7 AND NOT LL 9/9	Accepting Periodic Advertising Sync Transfer, Master Role, LE 2M	LL/CON/MAS/BV-96-C LL/CON/MAS/BV-98-C
LL 7/28 AND LL 9/9	Accepting Periodic Advertising Sync Transfer, Master Role, LE Coded	LL/CON/MAS/BV-95-C LL/CON/MAS/BV-97-C
LL 7/29	Receiving Long Control PDUs, Master Role, LE 1M	LL/CON/MAS/BV-105-C
LL 7/29 AND LL 9/7 AND NOT LL 9/9	Receiving Long Control PDUs, Master Role, LE 2M	LL/CON/MAS/BV-106-C
LL 7/29 AND LL 9/9	Receiving Long Control PDUs, Master Role, LE Coded	LL/CON/MAS/BV-107-C
LL 7/30	Receiving Long Control PDUs, Master Role, LE 1M	LL/CON/MAS/BV-108-C LL/CON/MAS/BV-111-C LL/CON/MAS/BV-114-C
LL 7/30 AND LL 9/7 AND NOT LL 9/9	Receiving Long Control PDUs, Master Role, LE 2M	LL/CON/MAS/BV-109-C LL/CON/MAS/BV-112-C LL/CON/MAS/BV-115-C
LL 7/30 AND LL 9/9	Receiving Long Control PDUs, Master Role, LE Coded	LL/CON/MAS/BV-110-C LL/CON/MAS/BV-113-C LL/CON/MAS/BV-116-C
LL 4/6a AND LL 4/8	Scanning for Periodic Advertising Periodic Sync Establishment Filtering Policies	LL/DDI/SCN/BV-34-C
LL 5/1	Connection Initiation Invalid CRC Connection Initiation Timeout	LL/CON/INI/BI-01-C LL/CON/INI/BV-04-C
TLL 5/1 AND LL 7/1 AND LL 7/2	Slave Packets Invalid CRC Slave Packets Invalid Address Connection Initiation Missed Replies Connection Initiation	LL/CON/INI/BI-02-C LL/ENC/INI/BI-01-C LL/CON/INI/BV-03-C LL/CON/INI/BV-01-C
LL 5/1 AND LL 5/3 AND LL 7/1 AND LL 7/2	Initiation Device Filtering: Undirected Initiation Device Filtering: Directed	LL/CON/INI/BV-06-C LL/CON/INI/BV-07-C
LL 5/1 AND LL 8/3 AND NOT LL 9/10	Requesting Connections, Support Data channel selection algorithm	LL/CON/INI/BV-22-C
LL 5/2 AND LL 7/1 AND LL 7/2	Connecting to Directed Advertising	LL/CON/INI/BV-02-C



Item	Feature	Test Case(s)
LL 5/1 AND LL 2/4 AND LL 2/5	Privacy – Connection Establishment	LL/CON/INI/BV-08-C LL/CON/INI/BV-09-C
LL 5/1 AND LL 2/4 AND LL 2/6	Generation of private addresses, Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Connection Establishment	LL/CON/INI/BV-18-C
LL 5/1 AND LL 2/4 AND LL 2/7	Generation of private addresses, Device Privacy Mode, Connection Establishment	LL/CON/INI/BV-20-C
LL 5/2 AND LL 2/4 AND LL 2/5	Privacy – Connection Establishment with directed advertisement	LL/CON/INI/BV-10-C LL/CON/INI/BV-11-C LL/CON/INI/BV-12-C
LL 5/2 AND LL 2/4 AND LL 2/6	Generation of private addresses, Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Connection Establishment with directed advertisement	LL/CON/INI/BV-19-C
LL 5/2 AND LL 2/4 AND LL 2/7	Generation of private addresses, Device Privacy Mode, Connection Establishment with directed advertisement	LL/CON/INI/BV-21-C
LL 5/1 AND LL 2/6	Network Privacy - Connection Establishment using whitelist and resolving list with address resolution disabled	LL/CON/INI/BV-23-C
	Network Privacy - Connection Establishment using resolving list with address resolution disabled	LL/CON/INI/BV-24-C
LL 4/7 AND LL 5/1	Extended Scanning, Connection Initiation	LL/CON/INI/BV-13-C
LL 5/1 AND LL 9/10	Requesting Connections, Channel Selection Algorithm #2	LL/CON/INI/BV-14-C LL/CON/INI/BV-15-C LL/CON/INI/BV-16-C LL/CON/INI/BV-17-C
LL 4/1 AND LL 4/6 AND LL 2/5	Privacy – Passive Scanning	LL/DDI/SCN/BV-13-C LL/DDI/SCN/BV-14-C
LL 4/1 AND LL 2/6	Network Privacy Mode – Ignore Identity Address when IRK is present in resolving list, Passive Scanning	LL/DDI/SCN/BV-26-C
LL 4/1 AND LL 2/7	Device Privacy Mode, Passive Scanning	LL/DDI/SCN/BV-28-C
LL 4/3 AND LL 2/4 AND LL 2/5	Privacy – Active Scanning	LL/DDI/SCN/BV-15-C LL/DDI/SCN/BV-16-C LL/DDI/SCN/BV-17-C LL/DDI/SCN/BV-18-C LL/DDI/SCN/BI-03-C
LL 3/2 AND LL 3/6		LL/TIM/ADV/BV-01-C LL/TIM/ADV/BV-02-C



Item	Feature	Test Case(s)
LL 6/1		LL/ENC/SLA/BI-01-C LL/TIM/SLA/BV-02-C LL/TIM/SLA/BV-03-C LL/TIM/SLA/BV-04-C
LL 6/1 AND LL 6/2		LL/CON/SLA/BI-01-C LL/CON/SLA/BV-02-C LL/CON/SLA/BV-15-C LL/CON/SLA/BV-16-C LL/CON/SLA/BV-17-C LL/CON/SLA/BV-18-C LL/TIM/SLA/BV-05-C
LL 6/1 AND LL 6/2 AND LL 6/19		LL/CON/SLA/BV-19-C LL/CON/SLA/BV-20-C
LL 6/1 AND LL 6/2		LL/CON/SLA/BV-21-C
LL 6/14		LL/CON/SLA/BI-02-C
LL 6/1 AND LL 6/2 AND LL 6/10 AND LL 6/11		LL/CON/SLA/BI-04-C
LL 6/1 AND LL 6/2 AND LL 6/10		LL/CON/SLA/BV-10-C LL/TIM/SLA/BV-01-C
LL 6/1 AND LL 6/2 AND LL 6/11		LL/FRH/SLA/BV-01-C
LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/13 AND LL 6/5 AND LL 6/6		LL/SEC/SLA/BI-01-C
LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/5 AND LL 6/6		LL/SEC/SLA/BV-01-C LL/SEC/SLA/BV-04-C
LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/5 AND LL 6/6 AND LL 6/18		LL/SEC/SLA/BV-02-C LL/SEC/SLA/BV-03-C
LL 6/1 AND LL 6/7 AND LL 6/20		LL/TIM/SLA/BV-06-C LL/TIM/SLA/BV-07-C
LL 6/1 AND LL 6/2 AND LL 6/13		LL/CON/SLA/BI-05-C
LL 6/1 AND LL 6/2 AND LL 6/14		LL/CON/SLA/BV-11-C
LL 6/1 AND LL 6/2 AND LL 6/15		LL/CON/SLA/BV-12-C
LL 6/1 AND LL 6/2 AND LL 6/16		LL/CON/SLA/BV-13-C
LL 6/1 AND LL 6/2 AND LL 6/3		LL/PAC/SLA/BV-01-C LL/PAC/SLA/BI-01-C
LL 6/1 AND LL 6/2 AND LL 6/4		LL/CON/SLA/BV-14-C
LL 6/1 AND LL 6/2 AND LL 6/5	Slave Transmitting Fragmented L2CAP Header	LL/CON/SLA/BV-04-C LL/DFL/SLA/BV-01-C



Item	Feature	Test Case(s)
LL 6/12		LL/SEC/SLA/BI-03-C LL/SEC/SLA/BI-04-C LL/SEC/SLA/BI-05-C
LL 6/1 AND LL 6/2 AND LL 6/6 AND LL 6/5		LL/CON/SLA/BV-06-C
LL 6/1 AND LL 6/2 AND LL 6/6	Slave Receiving Fragmented L2CAP Header	LL/CON/SLA/BV-05-C LL/DFL/SLA/BV-02-C
LL 6/1 AND LL 6/2 AND LL 6/11 AND LL 9/10	Slave Transmissions, Acknowledgement Scheme, Accepting Channel Map Update, Channel Selection Algorithm #2	LL/FRH/SLA/BV-02-C
LL 6/22 AND HCI 10/13	Slave Data Length Update	LL/CON/SLA/BV-77-C LL/CON/SLA/BV-78-C LL/CON/SLA/BV-79-C
LL 6/22 AND LL 9/7 AND HCI 10/13 AND HCI 13/12	Slave Data Length Update, LE 2M PHY	LL/CON/SLA/BV-80-C LL/CON/SLA/BV-81-C
LL 6/22 AND LL 9/9 AND HCI 10/13 AND HCI 13/12	Slave Data Length Update , LE Coded PHY	LL/CON/SLA/BV-82-C LL/CON/SLA/BV-83-C LL/CON/SLA/BV-84-C
LL 7/22 AND HCI 10/13	Master Data Length Update	LL/CON/MAS/BV-73-C LL/CON/MAS/BV-74-C LL/CON/MAS/BV-75-C
LL 7/22 AND LL 9/7 AND HCI 10/13 AND HCI 13/12	Master Data Length Update, LE 2M PHY	LL/CON/MAS/BV-76-C LL/CON/MAS/BV-77-C
LL 7/22 AND LL 9/9 AND HCI 10/13 AND HCI 13/12	Master Data Length Update, LE Coded PHY	LL/CON/MAS/BV-78-C LL/CON/MAS/BV-79-C LL/CON/MAS/BV-80-C
LL 6/24 AND LL 9/21	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)	LL/CON/SLA/BV-60-C LL/CON/SLA/BV-69-C LL/CON/SLA/BV-75-C
LL 6/24 AND LL 9/21 AND LL 6/12	Constant Tone Extension Request Procedure as Initiator, Slave Role,2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start	LL/CON/SLA/BV-71-C
LL 6/24 AND (LL 9/20 OR LL 9/21)	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD), 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)	LL/CON/SLA/BV-61-C LL/CON/SLA/BV-62-C LL/CON/SLA/BV-63-C LL/CON/SLA/BV-64-C



Item	Feature	Test Case(s)
LL 6/25 AND LL 9/19	Constant Tone Extension Request Procedure as Initiator, Slave Role, No Antenna Switching During Constant Tone Extension Transmission (AoA)	LL/CON/SLA/BV-65-C
LL 6/25 AND LL 9/19 AND LL 6/12	Constant Tone Extension Request Procedure as Initiator, Slave Role, No Antenna Switching During Constant Tone Extension Transmission (AoA) Encryption Start	LL/CON/SLA/BV-72-C
LL 6/25 AND (LL 9/18 OR LL 9/19)	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD), No Antenna Switching During Constant Tone Extension Transmission (AoA)	LL/CON/SLA/BV-66-C
LL 6/24 AND LL 9/20	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD)	LL/CON/SLA/BV-67-C LL/CON/SLA/BV-70-C LL/CON/SLA/BV-76-C
LL 6/24 AND LL 9/20 AND LL 6/12	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start	LL/CON/SLA/BV-73-C
LL 6/25 AND LL 9/18	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD)	LL/CON/SLA/BV-68-C
LL 6/25 AND LL 9/18 AND LL 6/12	Constant Tone Extension Request Procedure as Initiator, Slave Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD), Encryption Start	LL/CON/SLA/BV-74-C
LL 6/25	Constant Tone Extension Request Procedure as Responder, Slave Role	LL/CON/SLA/BV-87-C
LL 7/1		LL/TIM/MAS/BV-01-C LL/CON/MAS/BV-02-C



Item	Feature	Test Case(s)
LL 7/1 AND LL 7/2		LL/CON/MAS/BV-14-C LL/CON/MAS/BV-15-C LL/CON/MAS/BV-16-C LL/CON/MAS/BV-17-C LL/CON/MAS/BV-18-C LL/ENC/MAS/BI-01-C LL/TIM/MAS/BV-02-C
LL 7/1 AND LL 7/2 AND LL 7/10		LL/CON/MAS/BV-07-C
LL 7/1 AND LL 7/2 AND LL 7/11		LL/FRH/MAS/BV-01-C
LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6		LL/SEC/MAS/BV-01-C LL/SEC/MAS/BV-03-C
LL 7/1 AND LL 7/12 AND LL 7/6 AND LL 7/19		LL/SEC/MAS/BV-05-C
LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6 AND LL 7/19		LL/SEC/MAS/BV-02-C LL/SEC/MAS/BI-07-C
LL 7/1 AND LL 7/12 AND LL 7/6		LL/SEC/MAS/BV-04-C LL/SEC/MAS/BI-08-C LL/SEC/MAS/BI-09-C
LL 6/1 AND LL 6/2 AND LL 6/5 AND LL 6/6 AND LL 6/12		LL/SEC/SLA/BV-05-C
LL 7/1 AND LL 7/2 AND LL 7/13		LL/CON/MAS/BI-04-C
LL 7/1 AND LL 7/2 AND LL 7/14		LL/CON/MAS/BV-08-C
LL 7/1 AND LL 7/2 AND LL 7/15		LL/CON/MAS/BV-09-C
LL 7/1 AND LL 7/2 AND LL 7/16		LL/CON/MAS/BV-10-C
LL 7/1 AND LL 7/2 AND LL 7/3		LL/PAC/MAS/BV-01-C LL/PAC/MAS/BI-01-C
LL 7/1 AND LL 7/2 AND LL 7/4		LL/CON/MAS/BV-13-C
LL 7/1 AND LL 7/2 AND LL 7/4 AND LL 7/13		LL/CON/MAS/BV-19-C
LL 7/1 AND LL 7/2 AND LL 7/20		LL/CON/MAS/BV-20-C LL/CON/MAS/BV-21-C
LL 7/1 AND LL 7/2 AND LL 7/5 AND LL 7/6		LL/CON/MAS/BV-22-C
LL 7/14		LL/CON/MAS/BI-02-C



Item	Feature	Test Case(s)
LL 7/1 AND LL 7/2 AND LL 7/5 AND LL 7/6 AND LL 7/12 AND LL 7/16		LL/SEC/MAS/BI-01-C LL/SEC/MAS/BI-03-C LL/SEC/MAS/BI-04-C
LL 7/1 AND LL 7/2 AND LL 7/5	Master Transmitting Fragmented L2CAP Header	LL/CON/MAS/BV-03-C LL/DFL/MAS/BV-01-C
LL 7/1 AND LL 7/2 AND LL 7/6 AND LL 7/12		LL/SEC/MAS/BI-05-C LL/SEC/MAS/BI-06-C
LL 7/1 AND LL 7/2 AND LL 7/6	Master Receiving Fragmented L2CAP Header	LL/CON/MAS/BV-04-C LL/DFL/MAS/BV-02-C
LL 7/1 AND LL 7/2 AND LL 7/6 AND LL 7/5		LL/CON/MAS/BV-05-C
LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 9/10	Master Transmissions, Acknowledgement Scheme, Requesting Channel Map Update, Channel Selection Algorithm #2	LL/FRH/MAS/BV-02-C
LL 7/1 AND LL 7/23	Master Transmissions, Minimum Number Of Used Channels Procedure	LL/FRH/MAS/BV-03-C
LL 1/4 AND LL 9/2	LE Ping Procedure	LL/SEC/SLA/BV-06-C LL/SEC/SLA/BV-07-C LL/SEC/SLA/BV-08-C LL/SEC/SLA/BV-09-C LL/SEC/SLA/BV-10-C
LL 1/5 AND LL 9/2	LE Ping Procedure	LL/SEC/MAS/BV-06-C LL/SEC/MAS/BV-07-C LL/SEC/MAS/BV-08-C LL/SEC/MAS/BV-09-C LL/SEC/MAS/BV-10-C
LL 6/1 AND LL 6/2 AND LL 6/4a	Slave Initiated Feature Exchange	LL/CON/SLA/BV-22-C LL/CON/SLA/BV-23-C
LL 6/1 AND LL 6/2 AND LL 6/10a	Slave Initiating Connection Parameter Request	LL/CON/SLA/BV-24-C LL/CON/SLA/BV-25-C LL/CON/SLA/BV-26-C LL/CON/SLA/BV-27-C LL/CON/SLA/BV-85-C LL/CON/SLA/BV-86-C
LL 6/1 AND LL 6/2 AND LL 6/10a AND LL 6/13	Slave Initiating Connection Parameter Request – Timeout	LL/CON/SLA/BI-07-C
LL 6/1 AND LL 6/2 AND LL 6/10a AND LL 6/12	Slave Initiating Connection Parameter Request – Collision With Encryption Start	LL/CON/SLA/BV-28-C
LL 6/1 AND LL 6/2 AND LL 6/10b	Slave Accepting Connection Parameter Request	LL/CON/SLA/BV-29-C LL/CON/SLA/BV-30-C LL/CON/SLA/BV-31-C LL/CON/SLA/BV-32-C LL/CON/SLA/BV-33-C LL/CON/SLA/BI-08-C LL/CON/SLA/BV-34-C



Item	Feature	Test Case(s)
LL 7/1 AND LL 7/2 AND LL 7/4a	Master Responding to Feature Exchange	LL/CON/MAS/BV-23-C
LL 7/1 AND LL 7/2 AND LL 7/10a	Master Initiating Connection Parameter Request	LL/CON/MAS/BV-24-C LL/CON/MAS/BV-25-C LL/CON/MAS/BV-26-C LL/CON/MAS/BV-29-C LL/CON/MAS/BV-81-C LL/CON/MAS/BV-82-C
LL 7/1 AND LL 7/2 AND LL 7/10a AND LL 7/13	Master Initiating Connection Parameter Request – Timeout	LL/CON/MAS/BI-05-C
LL 7/1 AND LL 7/2 AND LL 7/10b	Master Accepting Connection Parameter Request	LL/CON/MAS/BV-30-C LL/CON/MAS/BV-31-C LL/CON/MAS/BV-32-C LL/CON/MAS/BV-33-C LL/CON/MAS/BV-34-C LL/CON/MAS/BI-06-C LL/CON/MAS/BV-27-C LL/CON/MAS/BV-35-C
LL 7/1 AND LL 7/2 AND LL 7/10b AND LL 7/12	Master Accepting Connection Parameter Request – Collision with Encryption Start	LL/CON/MAS/BV-28-C
LL 6/1 AND LL 6/2 AND LL 6/12 AND LL 6/5 AND LL 6/6	Slave Sending LL_REJECT_EXT_IND	LL/SEC/SLA/BV-11-C
LL 7/1 AND LL 7/2 AND LL 7/12 AND LL 7/5 AND LL 7/6 AND LL 9/4	Master Receiving LL_REJECT_EXT_IND	LL/SEC/MAS/BV-11-C
LL 7/12	Master Start Encryption: Overlapping Procedure; Unexpected Data Channel PDU	LL/SEC/MAS/BV-12-C LL/SEC/MAS/BV-14-C
LL 7/12 AND LL 9/5	Master Start Encryption: Overlapping Procedure	LL/SEC/MAS/BV-13-C
LL 7/24 AND LL 9/21	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)	LL/CON/MAS/BV-56-C LL/CON/MAS/BV-65-C LL/CON/MAS/BV-71-C
LL 7/24 AND LL 9/21 AND LL 7/12	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA) Encryption Start	LL/CON/MAS/BV-67-C
LL 7/24 AND (LL 9/20 OR LL 9/21)	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD), 2μs Antenna Switching And Sampling During Constant Tone Extension Reception (AoA)	LL/CON/MAS/BV-57-C LL/CON/MAS/BV-58-C LL/CON/MAS/BV-59-C LL/CON/MAS/BV-60-C



Item	Feature	Test Case(s)
LL 7/25 AND LL 9/19	Constant Tone Extension Request Procedure as Initiator, Master Role, No Antenna Switching During Constant Tone Extension Transmission (AoA)	LL/CON/MAS/BV-61-C
LL 7/25 AND LL 9/19 AND LL 7/12	Constant Tone Extension Request Procedure as Initiator, Master Role, No Antenna Switching During Constant Tone Extension Transmission (AoA) Encryption Start	LL/CON/MAS/BV-68-C
LL 7/25 AND (LL 9/18 OR LL 9/19)	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD), No Antenna Switching During Constant Tone Extension Transmission (AoA)	LL/CON/MAS/BV-62-C
LL 7/24 AND LL 9/20	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD)	LL/CON/MAS/BV-63-C LL/CON/MAS/BV-66-C LL/CON/MAS/BV-72-C
LL 7/24 AND LL 9/20 AND LL 7/12	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Sampling During Constant Tone Extension Reception (AoD), Encryption Start	LL/CON/MAS/BV-69-C
LL 7/25 AND LL 9/18	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD)	LL/CON/MAS/BV-64-C
LL 7/25 AND LL 9/18 AND LL 7/12	Constant Tone Extension Request Procedure as Initiator, Master Role, 2μs Antenna Switching During Constant Tone Extension Transmission (AoD), Encryption Start	LL/CON/MAS/BV-70-C
LL 7/25	Constant Tone Extension Request Procedure as Responder, Master Role	LL/CON/MAS/BV-83-C
LL 1/1 AND LL 9/7	LE 2M PHY	LL/DDI/ADV/BV-20-C
LL 1/4 AND LL 9/6 AND LL 9/7	PHY Update Procedure (LE 2M PHY, Slave)	LL/CON/SLA/BV-49-C LL/CON/SLA/BV-50-C LL/CON/SLA/BV-52-C LL/CON/SLA/BV-53-C



Item	Feature	Test Case(s)
LL 1/4 AND LL 9/6a	PHY Update Procedure (multiple PHYs, Slave)	LL/CON/SLA/BV-44-C LL/CON/SLA/BV-45-C LL/CON/SLA/BV-48-C
LL 1/4 AND LL 6/23a	PHY Update Procedure (Slave)	LL/CON/SLA/BV-51-C LL/CON/SLA/BI-09-C
LL 1/4 AND LL 6/11 AND LL 9/6a	Channel Map Collision (Slave Role)	LL/CON/SLA/BV-46-C
LL 1/4 AND LL 6/10b AND LL 9/6a	Parameter Update Collision (Slave Role)	LL/CON/SLA/BV-47-C
LL 1/5 AND LL 9/6 AND LL 9/7	PHY Update Procedure (LE 2M PHY, Master)	LL/CON/MAS/BV-49-C LL/CON/MAS/BV-50-C
LL 1/5 AND LL 9/6a	PHY Update Procedure (Master)	LL/CON/MAS/BV-45-C LL/CON/MAS/BV-46-C LL/CON/MAS/BV-51-C
LL 1/5 AND LL 7/11 AND LL 7/23a	Channel Map Collision (Master Role)	LL/CON/MAS/BV-47-C
LL 1/5 AND LL 7/10b AND LL 7/23a	Parameter Update Collision (Master Role)	LL/CON/MAS/BV-48-C
LL 1/5 AND LL 9/6a AND NOT LL 9/8	PHY Update Procedure, Symmetric Links (Master)	LL/CON/MAS/BV-42-C LL/CON/MAS/BV-44-C
LL 1/5 AND LL 9/6a AND LL 9/8	PHY Update Procedure, Symmetric and Asymmetric Links (Master)	LL/CON/MAS/BV-41-C LL/CON/MAS/BV-43-C LL/CON/MAS/BV-117-C
LL 1/4 AND LL 6/23a	PHY Update Procedure (Slave)	LL/CON/SLA/BV-40-C
LL 1/4 AND LL 9/6a AND NOT LL 9/8	Responding to PHY Update Procedure, Symmetric Links (Slave)	LL/CON/SLA/BV-43-C
LL 1/4 AND LL 6/23a	Responding to PHY Update Procedure (Slave)	LL/CON/SLA/BV-42-C
LL 1/4 AND LL 9/9	PHY Update Procedure (LE Coded PHY, Slave)	LL/CON/SLA/BV-54-C LL/CON/SLA/BV-57-C
LL 1/4 AND LL 9/6 AND LL 9/9	PHY Update Procedure (LE Coded PHY, Data Length Extension, Slave)	LL/CON/SLA/BV-55-C LL/CON/SLA/BV-56-C LL/CON/SLA/BV-58-C LL/CON/SLA/BV-59-C
LL 1/5 AND LL 9/9	PHY Update Procedure (LE Coded PHY, Master)	LL/CON/MAS/BV-52-C LL/CON/MAS/BV-55-C
LL 1/5 AND LL 9/6 AND LL 9/9	PHY Update Procedure (LE Coded PHY, Data Length Extension, Master)	LL/CON/MAS/BV-53-C LL/CON/MAS/BV-54-C
LL 9/30 AND LL 6/31	Changeable Sleep Clock Accuracy, Slave	LL/TIM/SLA/BV-08-C LL/TIM/SLA/BV-10-C
LL 9/30	Sleep Clock Accuracy, Slave	LL/TIM/SLA/BV-09-C
LL 9/30 AND LL 7/31	Changeable Sleep Clock Accuracy, Master	LL/TIM/MAS/BV-03-C LL/TIM/MAS/BV-05-C LL/TIM/MAS/BV-06-C LL/TIM/MAS/BV-07-C



Item	Feature	Test Case(s)
LL 9/30	Sleep Clock Accuracy, Master	LL/TIM/MAS/BV-04-C

Table 5.1: Test Case Mapping

