

IPv6 READY

Phase-2 Conformance Test Specification
6LoWPAN

Technical Document

Revision 1.0.0b2



IPv6 Forum
BII (China), CHT-TL (Taiwan), IRISA (France)
UNH-IOL (USA)

<http://www.ipv6forum.org>
<http://www.ipv6ready.org>



MODIFICATION RECORD

Version	Date	Author	Change Log
Version 1.0.0a1	January 27, 2011	e. johnson	Merged test specifications for RFC 4944 and 6LoWPAN-hc
Version 1.0.0b1	November 5, 2012	t. winters	Merged test specification for 6LoWPAN-nd
Version 1.0.0b2	December 17, 2014	J. Corson	<ul style="list-style-type: none">• Adapted a new conformance test plan based on work done by the v6 ready group.• Added more comprehensive tests for IPHC. Hard to be specific because this test spec is completely redone from the ground up.• Added new test cases that deal with RFC 7400.• Ensured that the new test spec covers all test cases created in the original test spec.



ACKNOWLEDGEMENTS

The IPv6 Forum would like to acknowledge the efforts of the following organizations in the development of this test suite.

Principle Authors:

BII (China)
CHT-TL (Taiwan)
IRISA-INRIA
University of New Hampshire- InterOperability Lab (USA)
Yokogawa Electric Corporation

Commentators:

TAHI Project (Japan)



INTRODUCTION

Overview

The IPv6 forum plays a major role to bring together industrial actors, to develop and deploy the new generation of IP protocols. Contrary to IPv4, which started with a small closed group of implementers, the universality of IPv6 leads to a huge number of implementations. Interoperability has always been considered as a critical feature in the Internet community. Due to the large number of IPv6 implementations, it is important to give to the market a strong signal proving the interoperability degree of various products.

To avoid confusion in the mind of customers, a unique logo program has been defined. The IPv6 logo gives confidence to users that IPv6 is currently operational. It is also a clear indication that the technology will still be used in the future. This logo program contributes to the feeling that IPv6 is available and ready to be used.

Abbreviations and Acronyms

6CIO: 6LoWPAN Capabilities Indication Option.

6CO: 6Lowpan Context Option.

ABRO: Authoritative Border Router Option

ARO: Address Registration Option

CID: Context Identifier

DAD: Duplicate Address Detection

GHC: Generic Header Compression

HC: Header Compression

HUT: Host Under Test

IPHC: IP Header Compression

MPDU: MAC Protocol Data Unit

MSDU: MAC Service Data Unit

MTU: Maximum Transmission Unit

NA: Neighbor advertisement

NCE: Neighbor Cache Entry

NHC: Next Header Compression

NS: Neighbor solicitation

NUT: Node Under Test

PO: Prefix Option

RA: Router Advertisement

RS: Router Solicitation

RUT: Router Under Test

SLLA: Source Link-Layer Address

TLLA: Target Link-layer Address

TLLA: Target Link-layer Address

TN: Testing Node

TR: Testing Router



Equipment Type

6LoWPAN Node (LN): A 6LoWPAN Node is any host or router participating in the LoWPAN. This term is used when referring to situations in which either a host or router can play the role described.

6LoWPAN Router (LR): An intermediate router in the LoWPAN who can communicate with other 6LoWPAN Node in the same LoWPAN. 6LoWPAN routers are present only in route-over topologies.

6LoWPAN Border Router (LBR): A border router located at the junction of separate 6LoWPAN networks or between a 6LoWPAN network and another IP network. There may be one or more 6LBRs at the 6LoWPAN network boundary. A 6LBR is the responsible authority for IPv6 Prefix propagation for the 6LoWPAN network it is serving. An isolated LoWPAN also contains a 6LBR in the network, which provides the prefix(es) for the isolated network.

6LoWPAN Host: A host is participating in the LoWPAN who can communicate with other 6LoWPAN Node in the same LoWPAN.



Advanced Functionality Tests

The following tests may be omitted if the NUT does not support the advanced functionalities.

1. GHC
 - a. [Test 6LoWPAN.1.1.24: Host 6LoWPAN Capabilities Indication Option Formatting](#)
 - b. [Test 6LoWPAN.1.1.25: Router 6LoWPAN Capabilities Indication Option Formatting](#)
 - c. [Test 6LoWPAN.1.1.26: Neighbor Discovery 6LoWPAN Capabilities Indication Option Formatting](#)
 - d. [Test 6LoWPAN.2.1.14: LOWPAN_GHC IPv6 Extension Header Encoding](#)
 - e. [Test 6LoWPAN.2.1.15: LOWPAN_GHC Compressed UDP Encoding](#)
 - f. [Test 6LoWPAN.2.1.16: LOWPAN_GHC Compressed ICMPv6 Encoding](#)
 - g. [Test 6LoWPAN.3.1.12: 6LoWPAN Capabilities Indication Option Formatting](#)
 - h. [Test 6LoWPAN.3.2.12: 6LoWPAN Capabilities Indication Option Formatting](#)



TEST ORGANIZATION

This document organizes tests by group based on related test methodology or goals. Each group begins with a brief set of comments pertaining to all tests within that group. This is followed by a series of description blocks; each block describes a single test. The format of the description block is as follows:

- Test Label:** The Test Label and Title comprise the first line of the test block. The Test Label is composed of the short test suite name, the group number, and the test number within the group, separated by periods.
- Purpose:** The Purpose is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
- References:** The References section lists cross-references to the specifications and documentation that might be helpful in understanding and evaluating the test and results.
- Test Setup:** The Test Setup section describes the configuration of all devices prior to the start of the test. Different parts of the procedure may involve configuration steps that deviate from what is given in the test setup. If a value is not provided for a protocol parameter, then the protocol's default is used.
- Procedure:** This section of the test description contains the step-by-step instructions for carrying out the test. These steps include such things as enabling interfaces, unplugging devices from the network, or sending packets from a test station. The test procedure also cues the tester to make observations, which are interpreted in accordance with the observable results given for that test part.
- Observable Results:** This section lists observable results that can be examined by the tester to verify that the NUT is operating properly. When multiple observable results are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail for each test is usually based on how the NUT's behavior compares to the results described in this section.
- Possible Problems:** This section contains a description of known issues with the test procedure, which may affect test results in certain situations.



REFERENCES

The following documents are referenced in these texts:

- [RFC 4944] Montenegro, G., N. Kushalnager, J. Hui, D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4, RFC 4944, September 2007.
- [RFC 6775] Shelby, Z., S. Chakrabarti, E. Nordmark, C. Bormann, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, November 2012.
- [RFC 6282] Hui, J., P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, RFC 6282, September 2011.
- [RFC 2460] Hinden, R., S. Deering, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.
- [RFC 4861] Narten, T., Nordmark, E., and W. Simpson, H. Soliman, Neighbor Discovery for IP Version 6 (IPv6), RFC 4861, September 2007.
- [RFC 4862] Thomson, S., T. Narten, T. Jinmei, IPv6 Stateless Address Autoconfiguration, RFC 4862, September 2007.
- [RFC 4443] Conta, A., S. Deering M. Gupta, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 4443, March 2006.
- [RFC 7400] C. Bormann, Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC7400, November 2014.







TABLE OF CONTENTS

MODIFICATION RECORD	2
ACKNOWLEDGEMENTS	2
INTRODUCTION	3
TEST ORGANIZATION	6
REFERENCES	7
TABLE OF CONTENTS	8
<i>Common Topology</i>	10
<i>Common Mesh Topology</i>	10
<i>Common Test Setup</i>	11
<i>Common Mesh Test Setup</i>	11
<i>Common Test Cleanup (for all tests)</i>	11
Section 1: RFC 2460	12
Section 2: RFC 4443	13
Section 3: RFC 4944	14
GROUP 1: FRAME FORMAT	15
Test 6LoWPAN.1.1.1: NALP Datagram Formatting.....	16
Test 6LoWPAN.1.1.2: Uncompressed IPv6 Header Formatting.....	19
Test 6LoWPAN.1.1.3: LOWPAN_HC1 Compressed IPv6 Header Formatting.....	21
Test 6LoWPAN.1.1.4: LOWPAN_HC1 Compressed UDP Formatting.....	24
Test 6LoWPAN.1.1.5: LOWPAN_BC0 Header Formatting.....	27
Test 6LoWPAN.1.1.6: Mesh Header Frame Formatting.....	29
Test 6LoWPAN.1.1.7: Fragmentation Frame Formatting.....	31
Test 6LoWPAN.1.1.8: Reserved Dispatch Pattern Formatting.....	34
Test 6LoWPAN.1.1.9: EUI-64 Address Formatting.....	36
Test 6LoWPAN.1.1.10: 16-bit Short Address Formatting.....	38
Test 6LoWPAN.1.1.11: Stateless Address Autoconfiguration Address Formatting.....	41
GROUP 2: HEADER COMPRESSION OPERATIONS	43
Test 6LoWPAN.2.1.1: LOWPAN_HC1 Address Encoding.....	44
Test 6LoWPAN.2.1.2: LOWPAN_HC1 Traffic Class and Flow Label Encoding.....	46
Test 6LoWPAN.2.1.3: LOWPAN_HC1 Compressed UDP Encoding.....	48
Section 4: RFC 6282	50
GROUP 1: FRAME FORMAT	51
Test 6LoWPAN.1.1.1: LOWPAN_IPHC Header Formatting.....	52
Test 6LoWPAN.1.1.2: LOWPAN_IPHC Context Identifier Field Formatting.....	54
Test 6LoWPAN.1.1.3: LOWPAN_IPHC Source Address Field Formatting.....	56
Test 6LoWPAN.1.1.4: LOWPAN_IPHC Non-Multicast Destination Address Field Formatting.....	59
Test 6LoWPAN.1.1.5: LOWPAN_IPHC Multicast Destination Address Field Formatting.....	62
Test 6LoWPAN.1.1.6: LOWPAN_IPHC NHC on the Extension Header Formatting.....	66



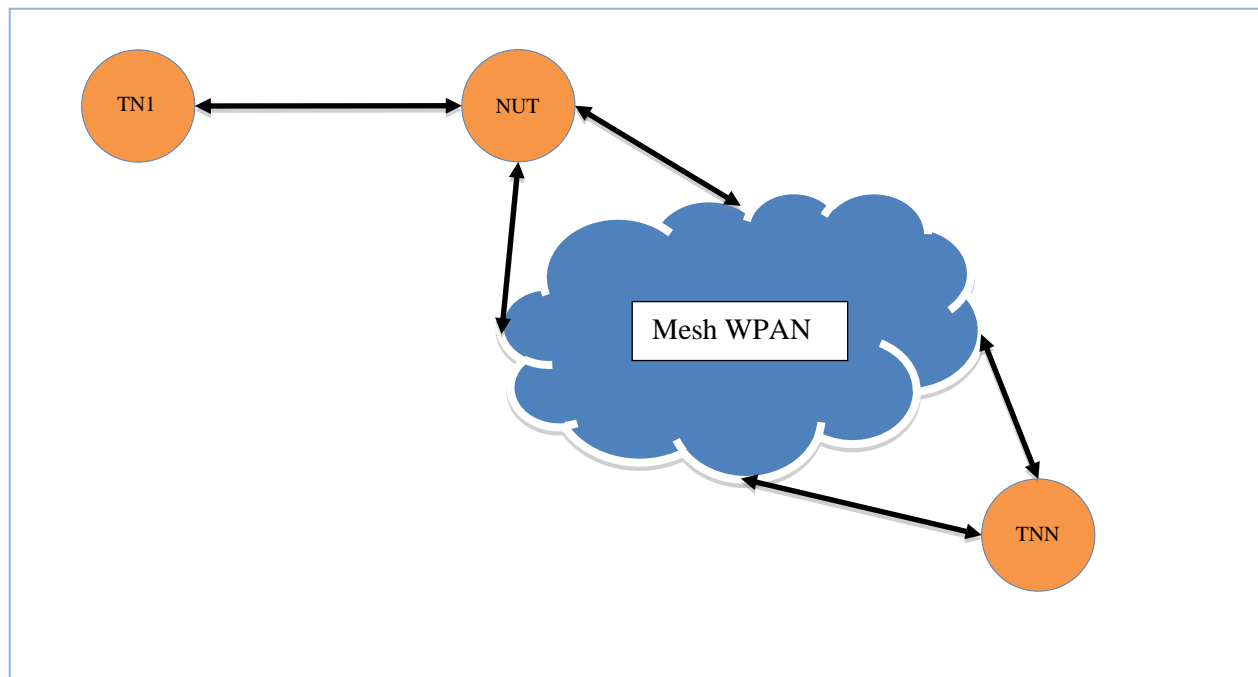
Test 6LoWPAN.1.1.7: LOWPAN_IPHC NHC UDP Formatting	68
Test 6LoWPAN.1.1.8: LOWPAN_IPHC Reserved NHC Values Formatting.....	71
Test 6LoWPAN.1.1.9: ESC Header Frame Formatting.....	73
GROUP 2: HEADER COMPRESSION OPERATIONS	75
Test 6LoWPAN.2.1.1: LOWPAN_IPHC Traffic Class and Flow Label Encoding.....	76
Test 6LoWPAN.2.1.2: LOWPAN_IPHC Next Header Compression	78
Test 6LoWPAN.2.1.3: LOWPAN_IPHC Hop Limit Encoding	79
Test 6LoWPAN.2.1.4: LOWPAN_IPHC Context Identifier Encoding	81
Test 6LoWPAN.2.1.5: LOWPAN_IPHC Source Address Compression Encoding	84
Test 6LoWPAN.2.1.6: LOWPAN_IPHC Non-Multicast Destination Address Compression Encoding.....	86
Test 6LoWPAN.2.1.7: LOWPAN_IPHC Multicast Destination Address Compression Encoding In Context Equal to 0.....	88
Test 6LoWPAN.2.1.8: LOWPAN_IPHC Multicast Destination Address Compression Encoding In Context Not Equal to 0.....	91
Test 6LoWPAN.2.1.9: LOWPAN_IPHC IPv6 Extension Header Encoding	93
Test 6LoWPAN.2.1.10: LOWPAN_IPHC Compressed UDP Encoding.....	95
Section 5: RFC 6775.....	97
GROUP 1: FRAME FORMAT	98
Test 6LoWPAN.1.1.1: Duplicate Address Request Formatting.....	99
Test 6LoWPAN.1.1.2: Duplicate Address Confirmation Formatting	102
Test 6LoWPAN.1.1.3: Host Neighbor Advertisement Formatting.....	105
GROUP 2: NEIGHBOR DISCOVERY.....	108
Subsection 1: Host Behavior.....	109
Subsection 2: Router Behavior	130
Section 6: RFC 7400.....	150
GROUP 1: FRAME FORMAT	151
Test 6LoWPAN.1.1.1: Host 6LoWPAN Capabilities Indication Option Formatting	152
Test 6LoWPAN.1.1.2: Router 6LoWPAN Capabilities Indication Option Formatting.....	154
Test 6LoWPAN.1.1.3: Neighbor Discovery 6LoWPAN Capabilities Indication Option Formatting	156
GROUP 2: HEADER COMPRESSION OPERATIONS	159
Test 6LoWPAN.2.1.1: LOWPAN_GHC IPv6 Extension Header Encoding	159
Test 6LoWPAN.2.1.2: LOWPAN_GHC Compressed UDP Encoding	161
Test 6LoWPAN.2.1.3: LOWPAN_GHC Compressed ICMPv6 Encoding	163
GROUP 3: NEIGHBOR DISCOVERY.....	164
Subsection 1: Host Behavior.....	165
Subsection 2: Router Behavior	167

Common Topology

NUT	Pure	Hybrid
Host		
Router		



Common Mesh Topology





Common Test Setup

Tests in this test suite may refer to a common test setup procedure defined for this section. Unless otherwise stated in the test case. NUT basic requirements for this test suit are listed below:

1. Supports uncompressed IPv6 datagram.
2. IEEE802.15.4 settings:
 - i Channel, modulation, data-rate (Channels 11-26 at 2.4 GHz).
 - ii MAC mode is beaconless.
3. Security is off.
4. Install a CoAP Ping server on both Host and Router.
5. All ICMPv6 Echo Requests should have a 32 byte payload unless otherwise specified.
6. The following global prefix is used unless otherwise noted: 2001:6db::1

Common Mesh Test Setup

Tests in this test suite may refer to a common test setup procedure defined for this section. Unless otherwise stated in the test case. NUT basic requirements for this test suit are listed below:

1. Supports uncompressed IPv6 datagram.
2. IEEE802.15.4 settings:
 - i Channel, modulation, data-rate (Channels 11-26 at 2.4 GHz).
 - ii MAC mode is beaconless.
3. Security is off.
4. A minimum of two TNs.
5. All TNs, except TN1, must be connected together prior to test execution.
6. Install a CoAP Ping server on both the NUT and all TNs.
7. All ICMPv6 Echo Requests should have a 32 byte payload unless otherwise specified.
8. The following global prefix is used unless otherwise noted: 2001:6db::1

Common Test Cleanup (for all tests)

Summary: The Cleanup procedure should cause the NUT to transition Neighbor Cache entries created in this test to state INCOMPLETE and remove any entries from its Default Router and Prefix Lists. Disconnect NUT from all connected TNs.



Section 1: RFC 2460

Scope

The following tests cover the base specification for Internet Protocol, Version 6 (IPv6) Specification, Request For Comments 2460.

Overview

The tests in this group verify that a node properly

TODO: Copy over and adapt relevant tests from the existing conformance test plan so that it correctly tests 6LoWPAN.



Section 2: RFC 4443

Scope

The following tests cover the base specification for Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, Request For Comments 4443.

Overview

The tests in this group verify that a node properly

TODO: Copy over and adapt relevant tests from the existing conformance test plan so that it correctly tests 6LoWPAN.



Section 3: RFC 4944

Scope

The following tests cover the base specification for Transmission of IPv6 Packets over IEEE 802.15.4 Networks, Request For Comments 4944.

Overview

The tests in this group verify that a node properly processes the adaptation layer header, generates its interface identifier by 16-bit short address or EUI-64 and forms its link local address. This test also verifies the unicast address mapping when performing address resolution and multicast address mapping.



Group 1: Frame Format

Scope

The following tests cover the Dispatch Header, interface identifier generation, and reserved header fields.

Overview

The tests in this group verify the dispatch type and headers in LoWPAN Adaption layer.



Test 6LoWPAN.1.1.1: NALP Datagram Formatting

Purpose: To verify that the NUT can handle frames that contain the NALP dispatch pattern.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 00 ₂ followed by a valid uncompressed ICMPv6 Echo Request Frame with payload 10 bytes directed to NUT's EUI-64 link-local address.
MSDU2	LoWPAN dispatch pattern of 00 ₂ followed by a valid uncompressed ICMPv6 Echo Request Frame with payload 10 bytes directed to NUT's 16-bit short link-local address.
MSDU3	LoWPAN dispatch pattern of 00 ₂ followed by a valid compressed ICMPv6 Echo Request Frame with payload 10 bytes directed to NUT's EUI-64 link-local address.
MSDU4	LoWPAN dispatch pattern of 00 ₂ followed by a valid compressed ICMPv6 Echo Request Frame with payload 10 bytes directed to NUT's 16-bit short link-local address.
MSDU5	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 1: Test Frames

Procedure:

Part A: NALP Reception with EUI-64 Address uncompressed

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Disable header compression on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Part B: NALP Reception with 16-bit Short Address uncompressed

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Disable header compression on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Part C: NALP Reception with both EUI-64 address and 16-bit Short Addresses uncompressed

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Disable header compression on both TN1 and NUT.
4. Connect NUT to TN1.



5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.
9. Repeat steps 1-8 switching TN1 and NUT in steps 1 and 2. MSDU1 should be transmitted in step 5.

Part D: NALP Reception with EUI-64 Address compressed

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Enable header compression on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Part E: NALP Reception with 16-bit Short Address compressed

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Enable header compression on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU4 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Part F: NALP Reception with both EUI-64 address and 16-bit Short Addresses uncompressed

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Enable header compression on both TN1 and NUT.
4. Connect NUT to TN1.
5. TN1 transmits MSDU4 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.
9. Repeat steps 1-8 switching TN1 and NUT in steps 1 and 2. Use MSDU3 instead of MSDU4.

Observable Results:

Part A

Step 5: The NUT should silently discard MSDU1 and not forward an ICMPv6 Echo Reply to TN1.

Step 7: The NUT should respond to MSDU5 with an uncompressed ICMPv6 Echo Reply to ensure that is still connected.

Part B

Step 5: The NUT should silently discard MSDU2 and not forward an ICMPv6 Echo Reply to TN1.

Step 7: The NUT should respond to MSDU5 with an uncompressed ICMPv6 Echo Reply to ensure that is still connected.



Part C

Step 6: The NUT should silently discard MSDU3 and not forward an ICMPv6 Echo Reply to TN1.

Step 8: The NUT should respond to MSDU5 with an uncompressed ICMPv6 Echo Reply to ensure that is still connected.

Part D

Step 5: The NUT should silently discard MSDU3 and not forward an ICMPv6 Echo Reply to TN1.

Step 7: The NUT should respond to MSDU5 with a compressed ICMPv6 Echo Reply to ensure that is still connected.

Part E

Step 5: The NUT should silently discard MSDU4 and not forward an ICMPv6 Echo Reply to TN1.

Step 7: The NUT should respond to MSDU5 with a compressed ICMPv6 Echo Reply to ensure that is still connected.

Part F

Step 6: The NUT should silently discard MSDU3 and not forward an ICMPv6 Echo Reply to TN1.

Step 8: The NUT should respond to MSDU5 with a compressed ICMPv6 Echo Reply to ensure that is still connected.

Possible Problems:

- None.



Test 6LoWPAN.1.1.2: Uncompressed IPv6 Header Formatting

Purpose: To verify that the NUT can handle frames that contains the uncompressed IPv6 header dispatch.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 2460] – [3](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x41 followed by a valid IPv6 header containing an ICMPv6 Echo Request.

Table 2: Test Frames

Procedure:

Part A: Uncompressed IPv6 Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Part B: Uncompressed IPv6 Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Part C: Uncompressed IPv6 Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 1-5 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.



Possible Problems:

- None.



Test 6LoWPAN.1.1.3: LOWPAN_HC1 Compressed IPv6 Header Formatting

Purpose: To verify that the NUT can handle frames compressed using the HC1 compression format.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 4944] – [10.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing an ICMPv6 Echo Request. The IPv6 header shall be compressed using HC1. The ICMPv6 next header shall remain uncompressed.
MSDU2	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing an ICMPv6 Echo Request. The IPv6 header shall be compressed using HC1. The ICMPv6 next header will also be compressed using HC1.
MSDU3	LoWPAN dispatch pattern of 0x42 followed by 0x00. Following this there will be a valid IPv6 Header followed by an uncompressed ICMPv6 Echo Request.
MSDU4	LoWPAN dispatch pattern of 0x42 followed by 0x40. Following this there will be an IPv6 Header whose source address has the interface identifier elided followed by an uncompressed ICMPv6 Echo Request.
MSDU5	LoWPAN dispatch pattern of 0x42 followed by 0x80. Following this there will be an IPv6 Header whose source address has the link-local prefix elided followed by an uncompressed ICMPv6 Echo Request.
MSDU6	LoWPAN dispatch pattern of 0x42 followed by 0xC0. Following this there will be an IPv6 Header whose source address is elided followed by an uncompressed ICMPv6 Echo Request.
MSDU7	LoWPAN dispatch pattern of 0x42 followed by 0x10. Following this there will be an IPv6 Header whose destination address has the interface identifier elided followed by an uncompressed ICMPv6 Echo Request.
MSDU8	LoWPAN dispatch pattern of 0x42 followed by 0x20. Following this there will be an IPv6 Header whose destination address has the link-local prefix elided followed by an uncompressed ICMPv6 Echo Request.
MSDU9	LoWPAN dispatch pattern of 0x42 followed by 0x30. Following this there will be an IPv6 Header whose destination address is elided followed by an uncompressed ICMPv6 Echo Request.
MSDU10	LoWPAN dispatch pattern of 0x42 followed by 0x08. Following this there will be an IPv6 Header whose traffic class and flow label are elided followed by an uncompressed ICMPv6 Echo Request.
MSDU11	LoWPAN dispatch pattern of 0x42 followed by 0x040. Following this there will be an IPv6 Header whose destination address has the interface identifier followed by a compressed ICMPv6 Echo Request.



Table 3: Test Frames

Procedure:

Part A: LOWPAN_HC1 Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.

Part B: LOWPAN_HC1 Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.

Part C: LOWPAN_HC1 Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU2 to the NUT.
7. Observe NUT.
8. Repeat steps 1-7 where TN1 and the NUT are switched in step 1 and 2.

Part D: LOWPAN_HC1 encoding parsing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU3 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU4-11

Observable Results:

Part A

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.



Step 7: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part D

Step 4: The NUT should respond to all MSDU with an ICMPv6 Echo Reply. Ensure that the NUT's IPv6 field has the correct fields elided.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.4: LOWPAN_HC1 Compressed UDP Formatting

Purpose: To verify that the NUT can handle UDP frames compressed with HC1 compression format.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 4944] – [10.1](#)
- [RFC 4944] – [10.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing an ICMPv6 Echo Request. The IPv6 header shall be compressed using HC1. The ICMPv6 next header will also be compressed using HC1.
MSDU2	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using HC1. The UDP next header will also be compressed using HC1.
MSDU3	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using HC1. The IPv6 header shall be followed by 0x00. This is followed by an uncompressed source port field, destination port field, and length field. This will be followed by the UDP payload.
MSDU4	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using HC1. The IPv6 header shall be followed by 0x20. This is followed by an uncompressed source port field and destination port field. The length field will be compressed. This will be followed by the UDP payload.
MSDU5	LoWPAN dispatch pattern of 0x42 followed by a valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using HC1. The UDP header shall be properly compressed with HC1. Bits 3-7 of the header shall be set to 0x1F.

Table 4: Test Frames

Procedure:

Part A: LOWPAN_HC1 UDP Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure CoAP Ping server on port 61616 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Part B: LOWPAN_HC1 UDP Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure CoAP Ping server on port 61616 on both TN1 and NUT.



3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Part C: LOWPAN_HC1 UDP Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Configure CoAP Ping Server on port 61616 on both TN1 and NUT.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU2 to the NUT.
8. Observe NUT.
9. Repeat steps 1-8 where TN1 and the NUT are switched in step 1 and 2.

Part D: LOWPAN_HC1 UDP encoding parsing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure CoAP Ping Server on port 180 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.
8. Repeat steps 4-7 with MSDU4

Part E: LOWPAN_HC1 UDP Port Parsing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1's CoAP Ping Server on port 61616.
3. Configure the NUT's CoAP Ping Server on port 61616.
4. Connect NUT to TN1.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU1 to the NUT.
8. Observe NUT.
9. Repeat steps 2-8 with the following source port configurations. {61617, 61621, 61631, 61636}
10. Repeat step 9 as destination port configurations.

Part F: LOWPAN_HC1 UCP_HC encoding reserved fields

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure CoAP Ping server on port 61616 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU5 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.



Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 5: The NUT should respond to all MSDU with a CoAP Reset message. Ensure that the UDP_HC field is correctly formatted and that the correct fields of the header are elided.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 6: The NIT should respond to MSDU5 with a CoAP Reset message. Ensure that the source and destination ports are correctly compressed for the configured source and destination port.

Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part F

Step 5: The NUT should respond to MSDU5 with a CoAP Reset message. The NUT should ignore all set reserved fields.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.5: LOWPAN_BC0 Header Formatting

Purpose: To verify that the NUT can handle LOWPAN_BC0 frames.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x50 followed by 0x00
MSDU2	An ICMPv6 Echo Request directed to the NUT.

Table 5: Test Frames

Procedure:

Part A: LOWPAN_BC0 Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.

Part B: LOWPAN_BC0 Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.

Part C: LOWPAN_BC0 Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU2 to the NUT.
7. Observe NUT.
8. Repeat steps 1-7 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 4: The NUT should receive MSDU1 without failure.



Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should receive MSDU1 without failure.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should receive MSDU1 without failure.

Step 7: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.6: Mesh Header Frame Formatting

Purpose: To verify that the HUT can handle Mesh addressing and type frames.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)

Test Setup: The network is setup according to [Common Mesh Topology](#). [Common Mesh Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT under the Pure topology in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x81 followed by the EUI-64 address of TN1 and the EUI-64 address of the HUT. This frame will also include a compressed ICMPv6 Echo Request.
MSDU2	LoWPAN dispatch pattern of 0xB1 followed by the 16-bit short address of TN1 and the 16-bit short address of the HUT. This frame will also include a compressed ICMPv6 Echo Request.
MSDU3	LoWPAN dispatch pattern of 0xA1 followed by the EUI-64 address of TN1 and the 16-bit short address of the HUT. This frame will also include a compressed ICMPv6 Echo Request.
MSDU4	LoWPAN dispatch pattern of 0x91 followed by the EUI-64 address of TN1 and the 16-bit short address of the HUT. This frame will also include a compressed ICMPv6 Echo Request.
MSDU5	LoWPAN dispatch pattern of 0x8FF followed by the EUI-64 address of TN1 and the EUI-64 address of the TNN. This frame will also include a compressed ICMPv6 Echo Request.
MSDU6	An ICMPv6 Echo Request directed to the HUT.

Table 6: Test Frames

Procedure:

Part A: Mesh Reception with EUI-64 Address

1. Configure TN1 and HUT to use EUI-64 link-local Address.
2. Connect HUT to TN1.
3. TN1 transmits MSDU1 to the HUT.
4. Observe HUT.
5. TN1 transmits MSDU6 to the HUT.
6. Observe HUT.

Part B: Mesh Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the HUT.
2. Connect HUT to TN1.
3. TN1 transmits MSDU2 to the HUT.
4. Observe HUT.
5. TN1 transmits MSDU6 to the HUT.
6. Observe HUT.

Part C: Mesh Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the HUT.
2. Configure TN1 to use EUI-64 link-local Address.



3. Connect HUT to TN1.
4. TN1 transmits MSDU3 to the HUT.
5. Observe HUT.
6. TN1 transmits MSDU6 to the HUT.
7. Observe HUT.
8. Repeat steps 1-7 where TN1 and the HUT are switched in step 1 and 2. Change MSDU3 to MSDU4.

Part C: Mesh Reception with Deep Hop Field

1. Configure TN1, ..., TNN, and HUT to use EUI-64 link-local Address.
2. Connect HUT to TN1.
3. Connect HUT to Mesh WPAN.
4. TN1 transmits MSDU5 to the HUT.
5. Observe HUT.
6. TN1 transmits MSDU6 to the HUT.
7. Observe HUT.

Observable Results:

Part A

Step 4: The HUT should receive MSDU1 without failure.

Step 6: The HUT should respond to MSDU6 with an ICMPv6 Echo Reply.

Part B

Step 4: The HUT should receive MSDU2 without failure.

Step 6: The HUT should respond to MSDU6 with an ICMPv6 Echo Reply.

Part C

Step 5: The HUT should receive MSDU3 without failure.

Step 7: The HUT should respond to MSDU6 with an ICMPv6 Echo Reply.

Part D

Step 5: The HUT should receive without failure. The HUT should route the packet to TNN. TNN should forward an ICMPv6 Echo Reply through the HUT to TN1.

Step 7: The HUT should respond to MSDU6 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.7: Fragmentation Frame Formatting

Purpose: To verify that the NUT can handle the first fragment header and the subsequent fragment header.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0xC4001BA followed by the first 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload.
MSDU2	LoWPAN dispatch pattern of 0xE4001BAx followed 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload. X is the current offset into the payload.
MSDU3	LoWPAN dispatch pattern of 0xC1400BA followed by the first 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload.
MSDU4	LoWPAN dispatch pattern of 0xE1BAx followed by 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload. X is the current offset into the payload.
MSDU5	LoWPAN dispatch pattern of 0xC1BA followed by the first 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload.
MSDU6	LoWPAN dispatch pattern of 0xE1BAx followed by 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload. X is the current offset into the payload.
MSDU7	LoWPAN dispatch pattern of 0xC4001BA followed by the first 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload.
MSDU8	LoWPAN dispatch pattern of 0xE4001BA(x+2) followed 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload. X is the current offset into the payload.
MSDU9	LoWPAN dispatch pattern of 0xC4001BA followed by the first 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload.
MSDU10	LoWPAN dispatch pattern of 0xE4001BA(x-1) followed 32 bytes of a valid ICMPv6 Echo Request with a 1024byte payload. X is the current offset into the payload.

Table 7: Test Frames

Procedure:

Part A: Fragment Reception with data_size field with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part B: Fragment Reception with data_size field with 16-bit Short Address



1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part C: Fragment Reception with one data_size field with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU3 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU4 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part D: Fragment Reception with one data_size field with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU3 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU4 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part E: Fragment Reception without data_size field with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU5 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU6 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part F: Fragment Reception without data_size field with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU5 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU6 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part G: Fragment Reception with invalid data offset

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU7 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU8 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part H: Fragment Reception with overlapping data offset



1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU9 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU10 to the NUT.
6. Repeat step 5 till all of the fragments of the ICMPv6 Request have been sent.
7. Observe NUT

Part I: Fragment Timeout

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. Wait 30 seconds.
6. TN1 transmits MSDU2 to the NUT.
7. Repeat steps 5-6 till all of the fragments of the ICMPv6 Request have been sent.
8. Observe NUT

Observable Results:

Part A

Step 7: The NUT should respond to MSDU1 with a fragmented ICMPv6 Echo Reply.

Part B

Step 7: The NUT should respond to MSDU1 with a fragmented ICMPv6 Echo Reply.

Part C

Step 7: The NUT should respond to MSDU3 with a fragmented ICMPv6 Echo Reply.

Part D

Step 7: The NUT should respond to MSDU3 with a fragmented ICMPv6 Echo Reply.

Part E

Step 7: The NUT should respond to MSDU5 with a fragmented ICMPv6 Echo Reply.

Part F

Step 7: The NUT should respond to MSDU5 with a fragmented ICMPv6 Echo Reply.

Part G

Step 7: The NUT should not respond to MSDU6 with a fragmented ICMPv6 Echo Reply.

Part H

Step 7: The NUT should not respond to MSDU7 with a fragmented ICMPv6 Echo Reply.

Part I

Step 8: The NUT should not respond to MSDU1 with a fragmented ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.8: Reserved Dispatch Pattern Formatting

Purpose: To verify that the NUT can handle reserved dispatch pattern.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x42 followed by a valid ICMPv6 Echo Request.
MSDU2	LoWPAN dispatch pattern of 0x5F followed by a valid ICMPv6 Echo Request.
MSDU3	LoWPAN dispatch pattern of 0xC8 followed by a valid ICMPv6 Echo Request.
MSDU4	LoWPAN dispatch pattern of 0xDF followed by a valid ICMPv6 Echo Request.
MSDU5	LoWPAN dispatch pattern of 0xE8 followed by a valid ICMPv6 Echo Request.
MSDU6	LoWPAN dispatch pattern of 0xFF followed by a valid ICMPv6 Echo Request.
MSDU7	An ICMPv6 Echo Request.

Table 8: Test Frames

Procedure:

Part A: Reserved Dispatch Frame Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU7 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-6

Part B: Reserved Dispatch Frame Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-6

Part C: Reserved Dispatch Frame Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU2 to the NUT.



7. Observe NUT.
8. Repeat steps 4-7 with MSDU2-6
9. Repeat steps 1-8 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 4: The NUT should silent discard all test frames.

Step 6: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should silent discard all test frames.

Step 6: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should silent discard all test frames.

Step 7: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.9: EUI-64 Address Formatting

Purpose: To verify that the NUT properly formats the EUI-64 Link-Local Address and can handle an invalid prefix.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 4944] – [7](#)
- [RFC 4944] – [8](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An ICMPv6 Neighbor Advertisement with a valid source and destination address.
MSDU2	An ICMPv6 Neighbor Advertisement with an invalid source address type of 0x02.
MSDU3	An ICMPv6 Neighbor Advertisement with an invalid source addresses length of 0x03.
MSDU4	An ICMPv6 Neighbor Advertisement with a source address with pad bytes set.
MSDU5	A valid ICMPv6 Echo Request.

Table 9: Test Frames

Procedure:

Part A: EUI-64 Address Formatting.

1. Configure TN1 and the NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Wait for the NUT to send a Neighbor Solicit.
4. TN1 transmits MSDU1 to the NUT.
5. TN1 transmits MSDU5 to the NUT.
6. Observe NUT.

Part B: Invalid Address Type Parsing

1. Configure the NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Wait for the NUT to send a Neighbor Solicit.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Part C: Invalid Length Field

1. Configure NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Wait for the NUT to send a Neighbor Solicit.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.



7. Observe NUT.

Part D: Source Address with pad bytes set.

1. Configure NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Wait for the NUT to send a Neighbor Solicit.
4. TN1 transmits MSDU4 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU5 to the NUT.
7. Observe NUT.

Observable Results:

Part A

Step 6: The NUT should respond to MSDU5 with an ICMPv6 Echo Reply. The NUT should correctly format the source address with its EUI-64 address.

Part B

Step 5: The NUT should silently discard MSDU2.

Step 6: The NUT should respond to MSDU5 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should silently discard MSDU3.

Step 6: The NUT should respond to MSDU5 with an ICMPv6 Echo Reply.

Part D

Step 5: The NUT should silently discard MSDU4.

Step 6: The NUT should respond to MSDU5 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.10: 16-bit Short Address Formatting

Purpose: To verify that the NUT can handle different types of 16-bit short addresses.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 4944] – [9](#)
- [RFC 4944] – [12](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An ICMPv6 Neighbor Advertisement with a correct source address.
MSDU2	An ICMPv6 Neighbor Advertisement with an invalid source address type of 0x02.
MSDU3	An ICMPv6 Neighbor Advertisement with an invalid source address length of 0x02.
MSDU4	An ICMPv6 Neighbor Advertisement with a valid source address with pad bytes set.
MSDU5	An ICMPv6 Echo Request.

Table 10: Test Frames

Procedure:

Part A: Unicast 16-bit short address where the NUT uses a EUI-64 Address

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 for a unicast 16-bit short address.
3. Connect NUT to TN1.
4. Wait for the NUT to send a Neighbor Solicit.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.

Part B: Unicast 16-bit short address where the NUT uses a 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 for a unicast 16-bit short address.
3. Connect NUT to TN1.
4. Wait for the NUT to send a Neighbor Solicit.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.

Part C: Unicast 16-bit short address with an invalid type field

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 for a unicast 16-bit short address.
3. Connect NUT to TN1.
4. Wait for the NUT to send a Neighbor Solicit.



5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.

Part D: Unicast 16-bit short address with an invalid length field

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 for a unicast 16-bit short address.
3. Connect NUT to TN1.
4. Wait for the NUT to send a Neighbor Solicit.
5. TN1 transmits MSDU3 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.

Part E: Unicast 16-bit short address with pad bytes set

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 for a unicast 16-bit short address.
3. Connect NUT to TN1.
4. Wait for the NUT to send a Neighbor Solicit.
5. TN1 transmits MSDU4 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU5 to the NUT.
8. Observe NUT.

Part F: Reserved 16-bit short address

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 with address 0xA000 as its 16-bit short address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU5 to the NUT.
5. Observe NUT.
6. Repeat steps 2-5 for the following 16-bit short addresses: {0xC000, 0xE000}

Observable Results:

Part A

Step 6: The NUT should receive MSUD1 without failure.

Step 8: The NUT should respond to MSDU5 with an ICMPv6 Echo Reply.

Part B

Step 6: The NUT should receive MSUD1 without failure.

Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 6: The NUT should receive MSUD2 without failure.

Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 6: The NUT should receive MSUD3 without failure.



Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 6: The NUT should receive MSUD4 without failure.

Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part F

Step 5: The NUT should receive the frame without error.

Possible Problems:

- None.



Test 6LoWPAN.1.1.11: Stateless Address Autoconfiguration Address Formatting

Purpose: To verify that the NUT can correctly create a stateless address autoconfiguration address.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [6](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An ICMPv6 Echo Request.

Table 11: Test Frames

Procedure:

Part A: 16-bit short Address Autoconfigured Address Formatting Where PAN ID is Known

1. Configure NUT to use stateless addressing mode.
2. Configure PAN coordinator to issue the NUT a 16-bit short address.
3. Configure TN1 to use a EUI-64 link-local address.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part B: 16-bit short Address Autoconfigured Address Formatting Where PAN ID is Unknown

1. Configure the PAN coordinator to hide their PAN ID.
2. Configure NUT to use stateless addressing mode.
3. Configure PAN coordinator to issue the NUT a 16-bit short address.
4. Configure TN1 to use a EUI-64 link-local address.
5. Connect NUT to TN1.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Part C: EUI-64 Address Autoconfigured Address Formatting Where PAN ID is Known

1. Configure NUT to use stateless addressing mode.
2. Configure TN1 and the NUT to use a EUI-64 link-local address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part D: EUI-64 Address Autoconfigured Address Formatting Where PAN ID is Unknown

1. Configure the PAN coordinator to hide their PAN ID.
2. Configure NUT to use stateless addressing mode.
3. Configure PAN coordinator to issue the NUT a 16-bit short address.
4. Configure TN1 to use a EUI-64 link-local address.
5. Connect NUT to TN1.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.



Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT's source address should be correctly generated.

Part B

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT's source address should be correctly generated.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT's source address should be correctly generated.

Part D

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT's source address should be correctly generated.

Possible Problems:

- None.



Group 2: Header Compression Operations

Scope

The following tests cover the different compression mechanisms.

Overview

The tests in this group verify that the NUT is capable of correctly compressing 6LoWPAN packets using HC1 and NHC.



Test 6LoWPAN.2.1.1: LOWPAN_HC1 Address Encoding

Purpose: To verify that the NUT can correctly compress the address field.

References:

- [RFC 4944] – [10.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 12: Test Frames

Procedure:

Part A: HC1 Destination Address encoding with Differing Prefix with EUI-64 Addresses

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 to use EUI-64 link-local Address with a differing prefix than NUT. The address should be generated using the MAC address of TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Configure NUT to use EUI-64 link-local Address using the MAC address of the NUT.
7. Configure TN1 to use EUI-64 link-local Address with a differing prefix than NUT.
8. Connect NUT to TN1.
9. TN1 transmits MSDU1 to the NUT.
10. Observe NUT.

Part B: HC1 Destination Address encoding with Differing Prefix with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address. The NUT should receive a differing prefix than TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Part C: HC1 Destination Address encoding with Unknown Interface Identifier with EUI-64 Addresses

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 to use EUI-64 link-local Address with a differing prefix than NUT. The address should be generated using the MAC address of TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Configure NUT to use EUI-64 link-local Address using the MAC address of the NUT.
7. Configure TN1 to use EUI-64 link-local Address with a differing prefix than NUT.
8. Connect NUT to TN1.
9. TN1 transmits MSDU1 to the NUT.
10. Observe NUT.



Part D: HCI Destination Address encoding with Unknown Interface Identifier with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 00. The NUT should set bits 2 and 3 to 01. The source address should be carried inline as well as the prefix identifier for the destination address.

Step 10: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 01. The NUT should set bits 2 and 3 to 00. The source address prefix should be carried in line as well as the uncompressed destination address.

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 01. The NUT should set bits 2 and 3 to 01. The prefix for both source and destination address should be carried inline.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 10. The NUT should set bits 2 and 3 to 11. The source interface identifier should be carried inline.

Step 10: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 11. The NUT should set bits 2 and 3 to 10. The destination interface identifier should be carried inline.

Part D

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 0 and 1 to 11. The NUT should set bits 2 and 3 to 11. There should be no address field carried inline.

Possible Problems:

- None.



Test 6LoWPAN.2.1.2: LOWPAN_HC1 Traffic Class and Flow Label Encoding

Purpose: To verify that the NUT can correctly compress the traffic class and flow label fields.

References:

- [RFC 4944] – [10.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 13: Test Frames

Procedure:

Part A: HC1 Traffic Class encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Configure TN1 to tag all traffic as Default PHB
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: HC1 Traffic Class encoding with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. Configure TN1 to tag all traffic as Default PHB
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part C: HC1 Flow Label encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part D: HC1 Flow Label with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 4 set to 1. Traffic class and flow label should be carried inline.



Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 4 set to 1. Traffic class and flow label should be carried inline.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 4 set to 1. Traffic class and flow label should be carried inline.

Part D

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 4 set to 1. Traffic class and flow label should be carried inline.

Possible Problems:

- None.



Test 6LoWPAN.2.1.3: LOWPAN_HC1 Compressed UDP Encoding

Purpose: To verify that the NUT can correctly encode a UDP frame using HC1 encoding.

References:

- [RFC 4944] – [10.1](#)
- [RFC 4944] – [10.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using HC1. The UDP next header will also be compressed using HC1.

Table 14: Test Frames

Procedure:

Part A: LOWPAN_HC1 UDP Port Parsing with EUI-64 Addressing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1's CoAP Ping Server on port 61616.
3. Configure the NUT's CoAP Ping Server on port 61615.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. Repeat steps 2-6 with the following source port configurations. {61616, 61621, 61631, 61636}
8. Configure TN1's CoAP Ping Server on port 61615.
9. Configure the NUT's CoAP Ping Server on port 61616.
10. Connect NUT to TN1.
11. TN1 transmits MSDU1 to the NUT.
12. Observe NUT.
13. Repeat steps 8-12 with the following destination port configurations. {61616, 61621, 61631, 61636}

Part B: LOWPAN_HC1 UDP Port Parsing with 16-bit Short Addressing

1. Configure the PAN coordinator to issue 16-bit Short addresses to TN1 and the NUT.
2. Configure TN1's CoAP Ping Server on port 61616.
3. Configure the NUT's CoAP Ping Server on port 61615.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. Repeat steps 2-6 with the following source port configurations. {61616, 61621, 61631, 61636}
8. Configure TN1's CoAP Ping Server on port 61615.
9. Configure the NUT's CoAP Ping Server on port 61616.
10. Connect NUT to TN1.
11. TN1 transmits MSDU1 to the NUT.
12. Observe NUT.



- Repeat steps 8-12 with the following destination port configurations. {61616, 61621, 61631, 61636}

Observable Results:

Part A

Step 6: The NUT should respond to MSDU1 with a CoAP Reset message. Bits 5 and 6 of the HC1 header should be set to 01. The NUT should set bit 0 and 1 to 01_2 in the HC_UDP encoding header. The source port should be carried inline along with 4 bits to represent the destination port.

Step 12: The NUT should respond to MSDU1 with a CoAP Reset message. Bits 5 and 6 of the HC1 header should be set to 01. The NUT should set bits 0 and 1 to 10_2 in the HC_UDP encoding header. The source port has 4 bits carried inline along with an uncompressed destination port.

Part B

Step 6: The NUT should respond to MSDU1 with a CoAP Reset message. Bits 5 and 6 of the HC1 header should be set to 01. The NUT should set bit 0 and 1 to 01_2 in the HC_UDP encoding header. The source port should be carried inline along with 4 bits to represent the destination port.

Step 12: The NUT should respond to MSDU1 with a CoAP Reset message. Bits 5 and 6 of the HC1 header should be set to 01. The NUT should set bits 0 and 1 to 10_2 in the HC_UDP encoding header. The source port has 4 bits carried inline along with an uncompressed destination port.

Possible Problems:

- None.



Section 4: RFC 6282

Scope

The following tests cover the base specification for Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, Request For Comments 6282.

Overview

The tests in this group verify that a node properly compresses an IPv6 Packet using the IPHC header.



Group 1: Frame Format

Scope

The following tests cover the Dispatch Header, interface identifier generation, and reserved header fields.

Overview

The tests in this group verify the dispatch type and headers in LoWPAN Adaption layer.



Test 6LoWPAN.1.1.1: LOWPAN_IPHC Header Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC compression header. This test will also ensure that the NUT will correctly format the LOWPAN_IPHC compression header.

References:

- [RFC 6282] – [3.1.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An uncompressed ICMPv6 Echo Request with an IPv6 header compressed using IPHC.
MSDU2	An IPHC header set to 0x6377 followed by ECN, DSCP, 4-bit pad, and Flow Label fields. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPHC header set to 0x6B77 followed by ECN, 2-bit pad, and Flow Label fields. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPHC header set to 0x7377 followed by ECN and DSCP fields. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPHC header set to 0x7B77. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPHC header set to 0x7877 followed by a hop limit field of 5. The next header is an uncompressed ICMPv6 Echo Request.
MSDU7	An IPHC header set to 0x7977. The next header is an uncompressed ICMPv6 Echo Request.
MSDU8	An IPHC header set to 0x7A77. The next header is an uncompressed ICMPv6 Echo Request.

Table 15: Test Frames

Procedure:

Part A: LOWPAN_IPHC Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Part B: LOWPAN_IPHC Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.

Part C: LOWPAN_IPHC Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 1-5 where TN1 and the NUT are switched in step 1 and 2.

Part D: LOWPAN_IPHC Basic Field Compression



1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU2 to the NUT.
4. Observe NUT.
5. Repeat steps 3-4 for MSDU3-8.

Observable Results:

Part A

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part B

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part D

Step 4: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly the frame state of TN1 and should compress the same fields that TN1 did.

Possible Problems:

- None.



Test 6LoWPAN.1.1.2: LOWPAN_IPHC Context Identifier Field Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC context identifier compression header. This test will also ensure that the NUT will correctly format the LOWPAN_IPHC context identifier compression header.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0x00. The next header is an uncompressed ICMPv6 Echo Request.
MSDU2	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0x10. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0x01. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0xF0. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0x0F. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPHC header set to 0x7BF7 followed by a context identifier field set to 0xFF. The next header is an uncompressed ICMPv6 Echo Request.

Table 16: Test Frames

Procedure:

Part A: LOWPAN_IPHC Context Identifier Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 2-5 with MSDU2-6 set the context for TN1 and NUT based on the context of the current MSDU.

Part B: LOWPAN_IPHC Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure TN1 and NUT to use context 0 for global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 2-5 with MSDU2-6 set the context for TN1 and NUT based on the context of the current MSDU.



Part C: LOWPAN_IPHC Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Configure TN1 and the NUT to use context 0 for global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. Repeat steps 3-5 with MSDU2-6 set the context for TN1 and NUT based on the context of the current MSDU.
8. Repeat steps 1-7 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 5: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part B

Step 5: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part C

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Possible Problems:

- None.



Test 6LoWPAN.1.1.3: LOWPAN_IPHC Source Address Field Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC source address fields.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An IPHC header set to 0x7B07 followed by an uncompressed source address field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU2	An IPHC header set to 0x7B17 followed by the last 64-bits of the source address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPHC header set to 0x7B27 followed by the last 16-bits of the source address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPHC header set to 0x7B37. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPHC header set to 0x7B47. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPHC header set to 0x7BC7 followed by any bits of the IID that are not covered by the CID field and a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU7	An IPHC header set to 0x7BE7 followed by any bits of the IID that are not covered by the CID field and a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU8	An IPHC header set to 0x7BF7 followed by a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.

Table 17: Test Frames

Procedure:

Part A: LOWPAN_IPHC Stateless Source Context field Reception with EUI-64 Address and IPv6 Full Address

1. Configure the NUT to use EUI-64 link-local Address.
2. Configure TN1 to use a full 128-bit link-local IPv6 Address.
3. Configure TN1 and the NUT to use context 0.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part B: LOWPAN_IPHC Stateless Source Context field Reception with EUI-64 Address

1. Configure TN1 and the NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use context 0.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.

Part C: LOWPAN_IPHC Stateless Source Context field Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.



2. Configure TN1 and the NUT to use context 0.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.

Part D: LOWPAN_IPHC Stateless Source Context field Reception with EUI-64 Address and known Interface ID

1. Configure the NUT to use EUI-64 link-local Address.
2. Configure TN1 to use its MAC address as its EUI-64 link-local Address.
3. Configure TN1 and the NUT to use context 0.
4. Connect NUT to TN1.
5. TN1 transmits MSDU4 to the NUT.
6. Observe NUT.

Part E: LOWPAN_IPHC Stateful Source Context field Reception with EUI-64 Address and Unspecified Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU5 to the NUT.
5. Observe NUT.

Part F: LOWPAN_IPHC Stateful Source Context field Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU6 to the NUT.
5. Observe NUT.

Part G: LOWPAN_IPHC Stateful Source Context field Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU7 to the NUT.
5. Observe NUT.

Part H: LOWPAN_IPHC Stateful Source Context field Reception Where the Interface ID is included in the Encapsulation Header

1. Configure the NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Configure TN1 to use an address covered by the context and the encapsulation header.
4. Connect NUT to TN1.
5. TN1 transmits MSDU8 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part B

Step 5: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.



Part C

Step 6: The NUT should respond to MSDU3 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part D

Step 6: The NUT should respond to MSDU4 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part E

Step 5: The NUT should not respond to MSDU5.

Part F

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part G

Step 5: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part H

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Possible Problems:

- None.



Test 6LoWPAN.1.1.4: LOWPAN_IPHC Non-Multicast Destination Address Field Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC destination address fields when multicast is not used.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An IPHC header set to 0x7B70 followed by an uncompressed destination address field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU2	An IPHC header set to 0x7B71 followed by the last 16-bits of the destination address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPHC header set to 0x7B73. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPHC header set to 0x7B74. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPHC header set to 0x7BB5 followed by any bits of the IID that are not covered by the CID field and a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPHC header set to 0x7BB6 followed by any bits of the IID that are not covered by the CID field and a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU7	An IPHC header set to 0x7BB7 followed by a valid CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU8	A valid ICMPv6 Echo Request.

Table 18: Test Frames

Procedure:

Part A: LOWPAN_IPHC Stateless Destination Context field Reception with EUI-64 Address and IPv6 Full Address

1. Configure TN1 to use EUI-64 link-local Address.
2. Configure the NUT to use a full 128-bit link-local IPv6 Address.
3. Configure TN1 and the NUT to use context 0.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part B: LOWPAN_IPHC Stateless Destination Context field Reception with EUI-64 Address

1. Configure TN1 and the NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use context 0.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.

Part C: LOWPAN_IPHC Stateless Destination Context field Reception with 16-bit Short Address



1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure TN1 and the NUT to use context 0.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.

Part D: LOWPAN_IPHC Stateless Destination Context field Reception with EUI-64 Address and known Interface ID

1. Configure TN1 to use EUI-64 link-local Address.
2. Configure the NUT to use its MAC address as its EUI-64 link-local Address.
3. Configure TN1 and the NUT to use context 0.
4. Connect NUT to TN1.
5. TN1 transmits MSDU4 to the NUT.
6. Observe NUT.

Part E: LOWPAN_IPHC Stateful Destination Context field Reception with EUI-64 Address and Unspecified Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU5 to the NUT.
5. Observe NUT.

Part F: LOWPAN_IPHC Stateful Destination Context field Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU6 to the NUT.
5. Observe NUT.

Part G: LOWPAN_IPHC Stateful Destination Context field Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure TN1 and the NUT to use a non-zero context.
3. Connect NUT to TN1.
4. TN1 transmits MSDU4 to the NUT.
5. Observe NUT.

Part H: LOWPAN_IPHC Stateful Destination Context field Reception Where the Interface ID is included in the Encapsulation Header

1. Configure TN1 to use EUI-64 link-local Address.
2. Configure TN1 and the NUT to use a non-zero context.
3. Configure the NUT to use an address covered by the context and the encapsulation header.
4. Connect NUT to TN1.
5. TN1 transmits MSDU6 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part B



Step 5: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part C

Step 6: The NUT should respond to MSDU3 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part D

Step 6: The NUT should respond to MSDU4 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part E

Step 5: The NUT should not respond to MSDU5.

Part F

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part G

Step 5: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part H

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Possible Problems:

- None.



Test 6LoWPAN.1.1.5: LOWPAN_IPHC Multicast Destination Address Field Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC destination address fields when multicast is used.

References:

- [RFC 6282] – [3.1.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An IPHC header set to 0x7B78 followed by an uncompressed multicast address field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU2	An IPHC header set to 0x7B79 followed by 48-bits of the multicast address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPHC header set to 0x7B7A followed by 32-bits of the multicast address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPHC header set to 0x7B7B followed by 8-bits of the multicast address. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPHC header set to 0x7BBC followed by 48-bits of the multicast address, and a CID field. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPHC header set to 0x7B7D. The next header is an uncompressed ICMPv6 Echo Request.
MSDU7	An IPHC header set to 0x7B7E. The next header is an uncompressed ICMPv6 Echo Request.
MSDU8	An IPHC header set to 0x7B7F. The next header is an uncompressed ICMPv6 Echo Request.
MSDU9	A valid ICMPv6 Echo Request.

Table 19: Test Frames

Procedure:

Part A: IPHC 128-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:0203:0405:0607:0809:0010:2030:4050.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:0203:0405:0607:0809:0010:2030:4050.
6. Observe NUT.

Part B: IPHC 128-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:0203:0405:0607:0809:0010::2030:4050.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:0203:0405:0607:0809:0010::2030:4050.
6. Observe NUT.

Part C: IPHC 48-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.



2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01::0020:3040:5060.
4. Connect NUT to TN1.
5. NUT transmits MSDU2 to the multicast group ff01::0020:3040:5060.
6. Observe NUT.

Part D: IPHC 48-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01::0020:3040:5060.
4. Connect NUT to TN1.
5. NUT transmits MSDU2 to the multicast group ff01::0020:3040:5060.
6. Observe NUT.

Part E: IPHC 32-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020:3040.
4. Connect NUT to TN1.
5. NUT transmits MSDU3 to the multicast group ff01:::0020:3040.
6. Observe NUT.

Part F: IPHC 32-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020:3040.
4. Connect NUT to TN1.
5. NUT transmits MSDU3 to the multicast group ff01:::0020:3040.
6. Observe NUT.

Part G: IPHC 8-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020.
4. Connect NUT to TN1.
5. NUT transmits MSDU4 to the multicast group ff01:::0020.
6. Observe NUT.

Part H: IPHC 8-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020.
4. Connect NUT to TN1.
5. NUT transmits MSDU4 to the multicast group ff01:::0020.
6. Observe NUT.

Part I: LOWPAN_IPHC Unicast-Prefix-based Multicast Address.

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060.
Where P is the prefix and L is the nibbles used to encode the prefix itself.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060.
6. Observe NUT.



Part J: LOWPAN_IPHC Reserved Destination Address Field

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU5 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU8 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU6-7

Observable Results:

Part A

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part B

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part C

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part D

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part E

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part F

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part G

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part H

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header.

Part I

Step 6: The NUT should respond to all test frames with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part J

Step 4: The NUT should silently discard all test frames.



Step 6: The NUT should respond to MSDU8 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.6: LOWPAN_IPHC NHC on the Extension Header Formatting

Purpose: To verify that the NUT will correctly handle the LOWPAN_IPHC next header compression on the IPv6 Extension header.

References:

- [RFC 6282] – [4.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	An uncompressed ICMPv6 Echo Request with an IPv6 header compressed using IPHC.
MSDU2	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xE0. Following this there will be a hop-by-hop option header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU3	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xE2. Following this there will be a routing header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU4	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xE4. Following this there will be a fragment header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU5	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xE6. Following this there will be a Destination Options header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU6	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xE8. Following this there will be a Mobility header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU7	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xEA. The next header is an uncompressed ICMPv6 Echo Request.
MSDU8	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xEC. The next header is an uncompressed ICMPv6 Echo Request.
MSDU9	An IPv6 header compressed with IPHC indicating that the next header is using NHC. Following that will be 0xEE. Following this will be an uncompressed IPv6 Header. The next header is an uncompressed ICMPv6 Echo Request.
MSDU10	A Valid ICMPv6 Echo Request.

Table 20: Test Frames

Procedure:

Part A: LOWPAN_IPHC Extension Header Compression Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. Repeat steps 3-4 with MSDU2-6, MSDU9



Part B: LOWPAN_IPHC Extension Header Compression Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. Repeat steps 3-4 with MSDU2-6, MSDU9

Part C: LOWPAN_IPHC Extension Header Compression Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 4-5 with MSDU2-6, MSDU9
7. Repeat steps 1-6 where TN1 and the NUT are switched in step 1 and 2.

Part D: LOWPAN_IPHC Extension Header Reserved Field Formatting

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU7 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU10 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 for MSDU8.

Observable Results:

Part A

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part B

Step 4: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should correctly format the IPHC header based on the current network state.

Part D

Step 4: The NUT should silently discard all test frames.

Step 5: The NUT should respond to MSDU10 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.7: LOWPAN_IPHC NHC UDP Formatting

Purpose: To verify that the NUT can handle UDP frames compressed with IPHC NHC.

References:

- [RFC 6282] – [4.3](#)
- [RFC 6282] – [4.3.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	First is an IPv6 header shall be compressed using IPHC. The ICMPv6 next header will also be compressed using NHC.
MSDU2	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The UDP next header will also be compressed using NHC.
MSDU3	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The IPv6 header shall be followed by 0xF0. This is followed by an uncompressed checksum, source port, and destination port. This will be followed by the UDP payload.
MSDU4	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The IPv6 header shall be followed by 0xF4. This is followed by an uncompressed source port and destination port. This will be followed by the UDP payload.
MSDU5	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The IPv6 header shall be followed by 0xF5. This is followed by an uncompressed source port and the last 8-bits of the destination port. This will be followed by the UDP payload.
MSDU6	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The IPv6 header shall be followed by 0xF6. This is followed by an uncompressed destination port and the last 8-bits of the source address. This will be followed by the UDP payload.
MSDU7	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The IPv6 header shall be followed by 0xF7. This is followed by the last 4-bits of both the source port and destination port. This will be followed by the UDP payload.

Table 21: Test Frames

Procedure:

Part A: LOWPAN_IPHC UDP Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure CoAP Ping server on port 61616 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Part B: LOWPAN_HC1 UDP Reception with 16-bit Short Address



1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Configure CoAP Ping server on port 61616 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Part C: LOWPAN_IPHC UDP Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Configure CoAP Ping Server on port 61616 on both TN1 and NUT.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU2 to the NUT.
8. Observe NUT.
9. Repeat steps 1-8 where TN1 and the NUT are switched in step 1 and 2.

Part D: LOWPAN_IPHC UDP encoding parsing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure CoAP Ping Server on port 180 on both TN1 and NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.
8. Repeat steps 4-7 with MSDU4

Part E: LOWPAN_HC1 UDP Port Parsing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1's CoAP Ping Server on port 61616.
3. Configure the NUT's CoAP Ping Server on port 61616.
4. Connect NUT to TN1.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. TN1 transmits MSDU1 to the NUT.
8. Observe NUT.
9. Repeat steps 2-8 with the following source port configurations. {61617, 61621, 61631, 61636}
10. Repeat step 9 as destination port configurations.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.



Part C

Step 5: The NUT should respond to MSDU2 with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 5: The NUT should respond to all MSDU with a CoAP Reset message.

Step 7: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 6: The NIT should respond to MSDU5 with a CoAP Reset message.

Step 8: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.8: LOWPAN_IPHC Reserved NHC Values Formatting

Purpose: To verify that the NUT can handle reserved IPHC NHC patterns.

References:

- [RFC 6282] – [5](#)
- [RFC 7400] – [4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header compressed using IPHC followed by 0x00.
MSDU2	A valid IPv6 Header compressed using IPHC followed by 0xDA.
MSDU3	A valid IPv6 Header compressed using IPHC followed by 0xF8.
MSDU4	A valid IPv6 Header compressed using IPHC followed by 0xF9.
MSDU5	A valid IPv6 Header compressed using IPHC followed by 0xFA.
MSDU6	A valid IPv6 Header compressed using IPHC followed by 0xFB.
MSDU7	An ICMPv6 Echo Request.

Table 22: Test Frames

Procedure:

Part A: Reserved IPHC NHC Frame Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU7 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-6

Part B: Reserved IPHC NHC Frame Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-6

Part C: Reserved IPHC NHC Frame Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU2 to the NUT.



7. Observe NUT.
8. Repeat steps 4-7 with MSDU2-6
9. Repeat steps 1-8 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 4: The NUT should silent discard all test frames.

Step 6: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should silent discard all test frames.

Step 6: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should silent discard all test frames.

Step 7: The NUT should respond to MSDU7 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.9: ESC Header Frame Formatting

Purpose: To verify that the NUT can handle additional dispatch bits.

References:

- [RFC 4944] – [5](#)
- [RFC 4944] – [5.1](#)
- [RFC 6282] – [5](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	LoWPAN dispatch pattern of 0x70 followed by 0x00, followed by a valid ICMPv6 Echo Request.
MSDU2	LoWPAN dispatch pattern of 0x70 followed by 0x01 followed by a valid ICMPv6 Echo Request.
MSDU3	LoWPAN dispatch pattern of 0x70 followed by 0x02 followed by a valid ICMPv6 Echo Request.
MSDU4	LoWPAN dispatch pattern of 0x70 followed by 0x04 followed by a valid ICMPv6 Echo Request.
MSDU5	LoWPAN dispatch pattern of 0x70 followed by 0x06 followed by a valid ICMPv6 Echo Request.
MSDU6	LoWPAN dispatch pattern of 0x70 followed by 0x08 followed by a valid ICMPv6 Echo Request.
MSDU7	LoWPAN dispatch pattern of 0x70 followed by 0x0F followed by a valid ICMPv6 Echo Request.
MSDU8	LoWPAN dispatch pattern of 0x70 followed by 0x3F followed by a valid ICMPv6 Echo Request.
MSDU9	An ICMPv6 Echo Request directed to the NUT.

Table 23: Test Frames

Procedure:

Part A: LOWPAN_BC0 Reception with EUI-64 Address

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.
4. Observe NUT.
5. TN1 transmits MSDU9 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-9

Part B: LOWPAN_BC0 Reception with 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to both TN1 and the NUT.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to the NUT.



4. Observe NUT.
5. TN1 transmits MSDU2 to the NUT.
6. Observe NUT.
7. Repeat steps 3-6 with MSDU2-9

Part C: LOWPAN_BC0 Reception with both EUI-64 and 16-bit Short Address

1. Configure the PAN coordinator to issue 16-bit short addresses to the NUT.
2. Configure TN1 to use EUI-64 link-local Address.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. TN1 transmits MSDU2 to the NUT.
7. Observe NUT.
8. Repeat steps 4-7 with MSDU2-9
9. Repeat steps 1-8 where TN1 and the NUT are switched in step 1 and 2.

Observable Results:

Part A

Step 4: The NUT should receive MSDU1 without failure. The NUT should not send an ICMPv6 Echo Reply.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part B

Step 4: The NUT should receive MSDU1 without failure.

Step 6: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Part C

Step 5: The NUT should receive MSDU1 without failure.

Step 7: The NUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Group 2: Header Compression Operations

Scope

The following tests cover the different compression mechanisms.

Overview

The tests in this group verify that the NUT is capable of correctly compressing 6LoWPAN packets using IPHC and NHC.



Test 6LoWPAN.2.1.1: LOWPAN_IPHC Traffic Class and Flow Label Encoding

Purpose: To verify that the NUT can correctly compress the traffic class and flow label fields.

References:

- [RFC 6282] – [3.1.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 24: Test Frames

Procedure:

Part A: IPHC Traffic Class encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Configure TN1 to tag all traffic as Default PHB
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: IPHC Traffic Class encoding with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. Configure TN1 to tag all traffic as Default PHB
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part C: IPHC Flow Label encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part D: IPHC Flow Label with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part E: IPHC Traffic Class and Flow Label encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. Configure TN1 to tag all traffic with Default PHB.
5. TN1 transmits MSDU1 to the NUT.



6. Observe NUT.

Part F: IPHC Traffic Class and Flow Label with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Connect NUT to TN1.
3. Configure TN1 to send all traffic with a valid flow label.
4. Configure TN1 to tag all traffic with Default PHB.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 10. ECN and DSCP should be carried inline.

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 10. ECN and DSCP should be carried inline.

Part C

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 01. ECN and Flow Label should be carried inline. Ensure that no bits are set in the reserved pad bits.

Part D

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 01. ECN and Flow Label should be carried inline. Ensure that no bits are set in the reserved pad bits.

Part E

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 00. ECN, DSCP, and Flow Label should be carried inline. Ensure that no bits are set in the reserved pad bits.

Part F

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 3 and 4 set to 00. ECN, DSCP, and Flow Label should be carried inline. Ensure that no bits are set in the reserved pad bits.

Possible Problems:

- None.



Test 6LoWPAN.2.1.2: LOWPAN_IPHC Next Header Compression

Purpose: To verify that the NUT can correctly encode the next header.

References:

- [RFC 6282] - 4

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using IPHC. The UDP next header will be uncompressed.
MSDU2	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using IPHC. The UDP next header will also be compressed using IPHC_NHC.

Table 25: Test Frames

Procedure:

Part A: LOWPAN_IPHC NHC with EUI-64 Addressing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1 and the NUT's CoAP Ping ServerServer on port 61616.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: LOWPAN_IPHC NHC with 16-bit Short Addressing

1. Configure the PAN coordinator to issue 16-bit Short addresses to TN1 and the NUT.
2. Configure TN1 and the NUT's CoAP Ping ServerServer on port 61616.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 will be set to 0. The UDP header will be carried inline.

Part B

Step 5: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 will be set to 0. The UDP header will be carried inline.

Possible Problems:

- None.



Test 6LoWPAN.2.1.3: LOWPAN_IPHC Hop Limit Encoding

Purpose: To verify that the NUT can correctly compress the hop limit field.

References:

- [RFC 6282] – [3.1.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 26: Test Frames

Procedure:

Part A: IPHC Hop Limit encoding with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT with default hop limit to 1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 2-5 with the following hop limits: {64,255,10}

Part B: IPHC Hop Limit encoding with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and NUT with default hop limit to 1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.
6. Repeat steps 2-5 with the following hop limits: {64,255,10}

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

- For hop limit 1, the NUT should set bits 6 and 7 of the IPHC header to 01
- For hop limit 10, the NUT should set bits 6 and 7 of the IPHC header to 00. The hop limit should then be carried in line with a value of 10.
- For hop limit 64, the NUT should set bits 6 and 7 of the IPHC header to 10
- For hop limit 255, the NUT should set bits 6 and 7 of the IPHC header to 11

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply.

- For hop limit 1, the NUT should set bits 6 and 7 of the IPHC header to 01
- For hop limit 10, the NUT should set bits 6 and 7 of the IPHC header to 00. The hop limit should then be carried in line with a value of 10.
- For hop limit 64, the NUT should set bits 6 and 7 of the IPHC header to 10



- For hop limit 255, the NUT should set bits 6 and 7 of the IPHC header to 11

Possible Problems:

- None.



Test 6LoWPAN.2.1.4: LOWPAN_IPHC Context Identifier Encoding

Purpose: To verify that the NUT can correctly compress context identifier.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 27: Test Frames

Procedure:

Part A: IPHC Context Identifier encoding with source context non-zero with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure NUT to use a non-zero context for the global prefix.
3. Configure TN1 to use context 0 for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part B: IPHC Context Identifier encoding with source context non-zero with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure NUT to use a non-zero context for the global prefix.
3. Configure TN1 to use context 0 for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part C: IPHC Context Identifier encoding with destination context non-zero with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure NUT to use the 0 context for the global prefix.
3. Configure TN1 to use a non-zero context for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part D: IPHC Context Identifier encoding with destination context non-zero with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure NUT to use the 0 context for the global prefix.
3. Configure TN1 to use a non-zero context for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.



Part E: IPHC Context Identifier encoding with source and destination context non-zero with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure the NUT and TN1 to use a non-zero context for global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part F: IPHC Context Identifier encoding with source and destination context non-zero with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure the NUT and TN1 to use a non-zero context for global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Observable Results:

Part A

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to the NUT's configured context and destination context set to 0.

Part B

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to the NUT's configured context and destination context set to 0.

Part C

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to 0 and destination context set to TN1's configured context.

Part D

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to 0 and destination context set to TN1's configured context.

Part E

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to NUT's configured context and destination context set to TN1's configured context.

Part F

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bit 8 to 1. CID field should be carried in line with source context set to NUT's configured context and destination context set to TN1's configured context.



Possible Problems:

- None.



Test 6LoWPAN.2.1.5: LOWPAN_IPHC Source Address Compression Encoding

Purpose: To verify that the NUT can correctly compress the source address.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 28: Test Frames

Procedure:

Part A: IPHC Source Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: IPHC Source Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part C: IPHC Source Address encoding with context 0 with EUI-64 Address Generated from MAC.

1. Configure TN1 to use EUI-64 link-local Address.
2. Configure NUT to use EUI-64 link-local Address with its MAC address as the base.
3. Configure TN1 and the NUT to use context 0 for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part D: IPHC Source Address encoding with context non-zero with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part E: IPHC Context Identifier encoding with context non-zero with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.
3. Connect NUT to TN1.



4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part F: IPHC Source Address encoding with context non-zero context with EUI-64 Address Generated from MAC.

1. Configure TN1 to use EUI-64 link-local Address.
2. Configure NUT to use EUI-64 link-local Address with its MAC address as the base.
3. Configure TN1 and NUT to use a non-zero context for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 0001. The last 64-bits of the source address should be carried inline.

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 0010. The last 16-bits of the source address should be carried inline.

Part C

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 0011.

Part D

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 1101. The last 64-bits of the source address should be carried inline.

Part E

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 1110. The last 16-bits of the source address should be carried inline.

Part F

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 9, 10, and 11 to 1111.

Possible Problems:

- None.



Test 6LoWPAN.2.1.6: LOWPAN_IPHC Non-Multicast Destination Address Compression Encoding

Purpose: To verify that the NUT can correctly compress the non-multicast destination address.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the NUT

Table 29: Test Frames

Procedure:

Part A: IPHC Destination Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: IPHC Destination Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part C: IPHC Destination Address encoding with context 0 with EUI-64 Address Generated from MAC.

1. Configure NUT to use EUI-64 link-local Address.
2. Configure TN1 to use EUI-64 link-local Address with its MAC address as the base.
3. Configure TN1 and the NUT to use context 0 for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Part D: IPHC Destination Address encoding with context non-zero with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part E: IPHC Destination Address encoding with context non-zero with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.



3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part F: IPHC Destination Address encoding with context non-zero context with EUI-64 Address Generated from MAC.

1. Configure DUT to use EUI-64 link-local Address.
2. Configure TN1 to use EUI-64 link-local Address with its MAC address as the base.
3. Configure TN1 and NUT to use a non-zero context for the global prefix.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 00001. The last 64-bits of the destination address should be carried inline.

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 00010. The last 16-bits of the destination address should be carried inline.

Part C

Step 6: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 00011.

Part D

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 10101. The last 64-bits of the destination address should be carried inline.

Part E

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 10110. The last 16-bits of the destination address should be carried inline.

Part F

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply. The NUT should set bits 8, 12, 13, 14, and 15 to 10111.

Possible Problems:

- None.



Test 6LoWPAN.2.1.7: LOWPAN_IPHC Multicast Destination Address Compression Encoding In Context Equal to 0.

Purpose: To verify that the NUT can correctly compress the multicast destination address in a context equal to 0.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the multicast group that both TN1 and the NUT are part of.

Table 30: Test Frames

Procedure:

Part A: IPHC 128-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:0203:0405:0607:0809:0010:2030:4050.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:0203:0405:0607:0809:0010:2030:4050.
6. Observe NUT.
7. Observe TN1.

Part B: IPHC 128-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:0203:0405:0607:0809:0010::2030:4050.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:0203:0405:0607:0809:0010::2030:4050.
6. Observe NUT.
7. Observe TN1.

Part C: IPHC 48-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01::0020:3040:5060.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01::0020:3040:5060.
6. Observe NUT.
7. Observe TN1.

Part D: IPHC 48-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.



2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01::0020:3040:5060.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01::0020:3040:5060.
6. Observe NUT.
7. Observe TN1.

Part E: IPHC 32-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020:3040.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:::0020:3040.
6. Observe NUT.
7. Observe TN1

Part F: IPHC 32-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020:3040.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:::0020:3040.
6. Observe NUT.
7. Observe TN1.

Part G: IPHC 8-bit Multicast Address encoding with context 0 with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:::0020.
6. Observe NUT.
7. Observe TN1.

Part H: IPHC 8-bit Multicast Address encoding with context 0 with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use context 0 for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:::0020.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:::0020.
6. Observe NUT.
7. Observe TN1.

Observable Results:

Part A

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01000. The full multicast address is included inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B



Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01000. The full multicast address is included inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01001. The 48-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01001. The 48-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01010. The 32-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part F

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01010. The 32-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part G

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01011. The 8-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Part H

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 01011. The 8-bits of the multicast address will be carried inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.2.1.8: LOWPAN_IPHC Multicast Destination Address Compression Encoding In Context Not Equal to 0.

Purpose: To verify that the NUT can correctly compress the multicast destination address in a context not equal to 0.

References:

- [RFC 6282] – [3.1.1](#)
- [RFC 6282] – [3.1.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the multicast group that both TN1 and the NUT is part of.

Table 31: Test Frames

Procedure:

Part A: IPHC 48-bit Multicast Address encoding with a non-zero context with EUI-64 Addresses

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Configure TN1 and NUT to use a non-zero context for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060. Where P is the prefix and L is the nibbles used to encode the prefix itself.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060.
6. Observe NUT.
7. Observe TN1.

Part B: IPHC 48-bit Multicast Address encoding with a non-zero context with 16-bit short Addresses

1. Configure the PAN coordinator to issue NUT and TN1 a 16-bit short address.
2. Configure TN1 and the NUT to use a non-zero context for the global prefix.
3. Instruct TN1 and the NUT to join multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060. Where P is the prefix and L is the nibbles used to encode the prefix itself.
4. Connect NUT to TN1.
5. NUT transmits MSDU1 to the multicast group ff01:02LL:PPPP:PPPP:PPPP:0320:3040:5060.
6. Observe NUT.
7. Observe TN1.

Observable Results:

Part A

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 11100. 48-bits of the multicast address will be included inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.



Part B

Step 6: The NUT should format MSDU1 as follows: The NUT should set bits 8, 12, 13, 14, and 15 to 11100. 48-bits of the multicast address will be included inline.

Step 7: TN1 should respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.2.1.9: LOWPAN_IPHC IPv6 Extension Header Encoding

Purpose: To verify that the NUT can correctly compress the IPv6 extension headers.

References:

- [RFC 6282] – [4.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPHC compressed IPv6 packet with hop-by-hop option header.
MSDU2	A valid IPHC compressed IPv6 packet with router header.
MSDU3	A valid IPHC compressed IPv6 packet with fragmentation header.
MSDU4	A valid IPHC compressed IPv6 packet with destination option header.
MSDU5	A valid IPHC compressed IPv6 packet with mobility header.
MSDU6	A valid IPHC compressed IPv6 packet with another IPv6 Header followed by an ICMPv6 Echo Request.

Table 32: Test Frames

Procedure:

Part A: IPHC IPv6 Hop-by Hop Option Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU1 to NUT.
4. Observe NUT.

Part B: IPHC IPv6 Routing Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU2 to NUT.
4. Observe NUT.

Part C: IPHC IPv6 Fragment Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU3 to NUT.
4. Observe NUT.

Part D: IPHC IPv6 Destination Options Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU4 to NUT.
4. Observe NUT.

Part E: IPHC IPv6 Mobility Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU5 to NUT.



4. Observe NUT.

Part F: IPHC Additional IPv6 Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Connect NUT to TN1.
3. TN1 transmits MSDU6 to NUT.
4. Observe NUT.

Observable Results:

Part A

Step 4:

Part B

Step 4:

Part C

Step 4:

Part D

Step 4:

Part E

Step 4:

Part F

Step 4:

Possible Problems:

- None.



Test 6LoWPAN.2.1.10: LOWPAN_IPHC Compressed UDP Encoding

Purpose: To verify that the NUT can correctly encode a UDP frame using IPHC NHC encoding.

References:

- [RFC 6282] – [4.3](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using NHC. The UDP next header will also be compressed using NHC.

Table 33: Test Frames

Procedure:

Part A: LOWPAN_IPHC UDP Port compressing with EUI-64 Addressing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Configure TN1's CoAP Ping Server on port 61440.
3. Configure the NUT's CoAP Ping Server on port 61430.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. Repeat steps 2-6 with the following source port configurations. {61440,61445,61490,61470,61695}
8. Configure TN1's CoAP Ping Server on port 61430.
9. Configure the NUT's CoAP Ping Server on port 61440.
10. Connect NUT to TN1.
11. TN1 transmits MSDU1 to the NUT.
12. Observe NUT.
13. Repeat steps 8-12 with the following destination port configurations. {61440,61445,61490,61470,61695}

Part B: LOWPAN_IPHC UDP Port compressing with 16-bit Short Addressing

1. Configure the PAN coordinator to issue 16-bit Short addresses to TN1 and the NUT.
2. Configure TN1's CoAP Ping Server on port 61440.
3. Configure the NUT's CoAP Ping Server on port 61430.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.
7. Repeat steps 2-6 with the following source port configurations. {61440,61445,61490,61470,61695}
8. Configure TN1's CoAP Ping Server on port 61430.
9. Configure the NUT's CoAP Ping Server on port 61440.
10. Connect NUT to TN1.
11. TN1 transmits MSDU1 to the NUT.



12. Observe NUT.
13. Repeat steps 8-12 with the following destination port configurations. {61440,61445,61490,61470,61695}

Part C: LOWPAN_IPHC UDP Port uncompressed

1. Configure TN1 and the NUT to use EUI-64 addresses.
2. Configure TN1's CoAP Ping Server on port 61430.
3. Configure the NUT's CoAP Ping Server on port 61430.
4. Connect NUT to TN1.
5. TN1 transmits MSDU1 to the NUT.
6. Observe NUT.

Observable Results:

Part A

Step 6: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the NHC_UDP header should be set to 01. The source port and the last 8-bits of the destination port should be included inline.

Step 12: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the NHC_UDP header should be set to 01. The destination port and the last 8-bits of the source port should be included inline.

Part B

Step 6: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the NHC_UDP header should be set to 01. The source port and the last 8-bits of the destination port should be included inline.

Step 12: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the NHC_UDP header should be set to 10. The destination port and the last 8-bits of the source port should be included inline.

Part C

Step 6: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the NHC_UDP header should be set to 11. The source port and the destination port should be included inline.

Possible Problems:

- None.



Section 5: RFC 6775

Scope

The following tests cover the base specification for Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks, Request For Comments 6775.

Overview

The tests in this group verify that a node properly handle 6lowpan-nd.



Group 1: Frame Format

Scope

The following tests cover the Dispatch Header, interface identifier generation, and reserved header fields.

Overview

The tests in this group verify the dispatch type and headers in LoWPAN Adaption layer.



Test 6LoWPAN.1.1.1: Duplicate Address Request Formatting

Purpose: To verify that the RUT can properly handle the ICMPv6 Duplicate Address Request.

References:

- [RFC 6775] – [4.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT under the Pure topology in this test case.

Test Frames	
MSDU1	A valid ICMPv6 Echo Request.
MSDU2	A valid ICMPv6 Duplicate Address Request.
MSDU3	An ICMPv6 Duplicate Address Request with the EUI-64 address of the RUT.
MSDU4	An ICMPv6 Duplicate Address Request with code set to 0xff
MSDU5	An ICMPv6 Duplicate Address Request with Checksum of 0.
MSDU6	An ICMPv6 Duplicate Address Request with Reserved field set.
MSDU7	An ICMPv6 Duplicate Address Request with Registration lifetime set to 0.
MSDU8	An ICMPv6 Duplicate Address Request with status set to 0xff

Table 34: Test Frames

Procedure:

Part A: Valid Duplicate Address Request Reception with Unique Addresses

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU2 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part B: Valid Duplicate Address Request Reception with Non-Unique Addresses

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU3 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part C: Duplicate Address Request Reception with Non-zero code field

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU4 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part D: Duplicate Address Request Reception with Invalid Checksum

1. Configure TR1 and the RUT to use a EUI-64 link-local address.



2. Connect RUT to TR1.
3. TR1 transmits MSDU5 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part E: Duplicate Address Request Reception with Reserved Field Set

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU6 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part F: Duplicate Address Request Reception with Lifetime set to 0

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU2 to the RUT with a different unique EUI-64 address than was configured.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT with a source address equal to that of the one just registered.
6. Observe RUT.
7. TR1 transmits MSDU7 to the RUT with the registered EUI-64 address.
8. Observe RUT.
9. TR1 transmits MSDU1 to the RUT with a source address equal to that of the one just registered.
10. Observe RUT.

Part G: Duplicate Address Request Reception with Status Field Set.

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU6 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Observable Results:

Part A

Step 4: The RUT should respond to MSDU2 with an ICMPv6 Duplicate Address Confirmation with a status of 0.

Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 4: The RUT should respond to MSDU3 with an ICMPv6 Duplicate Address Confirmation with a status of 1.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 4: The RUT should respond to MSDU4 with an ICMPv6 Duplicate Address Confirmation with a status of 1.



Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 4: The RUT should not respond to MSDU5 with an ICMPv6 Duplicate Address Confirmation.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 4: The RUT should not respond to MSDU6 with an ICMPv6 Duplicate Address Confirmation.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part F

Step 4: The RUT should respond to MSDU2 with an ICMPv6 Duplicate Address Confirmation with a status of 0.

Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Step 8: The RUT should respond to MSDU7 with an ICMPv6 Duplicate Address Confirmation with a status of 0.

Step 10: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part G

Step 4: The RUT should receive MSDU8 without failure.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.2: Duplicate Address Confirmation Formatting

Purpose: To verify that the RUT can properly handle the ICMPv6 Duplicate Address Confirmation.

References:

- [RFC 6775] – [4.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT under the Pure topology in this test case.

Test Frames	
MSDU1	A valid ICMPv6 Echo Request.
MSDU2	A valid ICMPv6 Duplicate Address Confirmation.
MSDU3	An ICMPv6 Duplicate Address Confirmation with status set to 0x1.
MSDU4	An ICMPv6 Duplicate Address Confirmation with code set to 0xff.
MSDU5	An ICMPv6 Duplicate Address Confirmation with Checksum of 0.
MSDU6	An ICMPv6 Duplicate Address Confirmation with Registration lifetime set to 1.
MSDU7	An ICMPv6 Duplicate Address Confirmation with Reserved field set.

Table 35: Test Frames

Procedure:

Part A: Valid Duplicate Address Confirmation Reception

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. Observe RUT.
4. TR1 transmits MSDU2 to the RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part B: Valid Duplicate Address Confirmation Reception with Status 1

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. Observe RUT.
4. TR1 transmits MSDU3 to the RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part C: Duplicate Address Confirmation Reception with Non-zero code field

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU4 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part D: Duplicate Address Confirmation Reception with Invalid Checksum

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.



3. Observe RUT.
4. TR1 transmits MSDU5 to the RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part E: Duplicate Address Confirmation Reception with Reserved Field Set

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. TR1 transmits MSDU7 to the RUT.
4. Observe RUT.
5. TR1 transmits MSDU1 to the RUT.
6. Observe RUT.

Part F: Duplicate Address Confirmation Reception with Lifetime set to 1

1. Configure TR1 and the RUT to use a EUI-64 link-local address.
2. Connect RUT to TR1.
3. Observe RUT.
4. TR1 transmits MSDU6 to the RUT.
5. Observe RUT.
6. Wait 90 seconds.
7. TR1 transmits MSDU1 to the RUT.
8. Observe RUT.

Observable Results:

Part A

Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part F



Step 3: The RUT should send an ICMPv6 Duplicate Address Confirmation Message to TR1.

Step 6: The RUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.3: Host Neighbor Advertisement Formatting

Purpose: Verify that the HUT properly handles Neighbor Advertisements.

References:

- [RFC 6775] – [5.5.2](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT with the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request.
MSDU2	A valid Neighbor Advertisement with status code 0. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option length field set to 1.
MSDU3	A valid Neighbor Advertisement with status code 1. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with an Incorrect EUI-64 Address.
MSDU4	A valid Neighbor Advertisement with status code 3. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with status 0.

Table 36: Test Frames

Procedure:

Part A: Incorrect length

1. Connect HUT to TN1.
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1 to HUT.
4. Observe the HUT.

Part B: Incorrect EUI-64

1. Connect HUT to TN1.
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU3.
3. TN1 transmits MSDU1 to the HUT.
4. Observe the HUT.

Part C: Status indicating a success.

1. Initialize the network interface of the HUT
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU4
3. TN1 transmits MSDU1 to the HUT.
4. Observe the HUT.

Observable Results:

Part A

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.



Part B

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.





Group 2: Neighbor Discovery

Scope

The following tests cover the different Neighbor Discovery mechanisms.

Overview

The tests in this group verify that the NUT is capable of exhibiting the correct behavior when using the Neighbor Discovery protocol.



Subsection 1: Host Behavior

Scope

The following tests cover the behavior of the NUT acting as a host using Neighbor Discovery.

Overview

These tests ensure that the NUT can correctly use Neighbor Discovery Protocol when acting as a host in a network.



Test 6LoWPAN.2.1.1: Interface Initialization

Purpose: Verify that the HUT can properly form an EUI-64 address, sends Router Solicitations (RS), and join the All-nodes multicast group.

References:

- [RFC 6775] – [5.2](#)
- [RFC 4944] – [6](#)
- [RFC 4861] – [6.3.7](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the all-node multicast address

Table 37: Test Frames

Procedure:

Part A: Router Solicitation

1. Connect TN1 and the HUT.
2. Observe the HUT.

Part B: Join All-nodes address

1. Connect TN1 and the HUT.
2. TN1 transmits MSDU1
3. Observe the HUT.

Observable Results:

Part A

Step 2: The HUT sends a Router Solicitation with the following details:

- An IPv6 source address set to the HUT EUI-64 link-local address

Part B

Step 3: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply

Possible Problems:

- None.



Test 6LoWPAN.2.1.2: Sending a Router Solicitation

Purpose: Verify that the HUT can properly send and retransmits Router Solicitations when the router/prefix list is empty.

References:

- [RFC 6775] – [5.3](#)
- [RFC 4861] – [6.3.7](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid Router Advertisement with lifetime 30 seconds.
MSDU2	A valid Router Advertisement with lifetime 30 seconds. It also contains a Prefix Information Option with lifetime 30 seconds.

Table 38: Test Frames

Procedure:

Part A: Sending a Router Solicitation

1. Connect the HUT to TN1.
2. Observe the HUT.

Part B: Expiration of the default Router

1. Connect the HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU1.
3. Wait for 30 seconds
4. Observe the HUT

Part C: Expiration of the global prefix

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. Wait for 30 seconds
4. Observe the HUT

Part D: Retransmission of Router Solicitations

1. Initialize the network interface of the HUT.
2. Observe the HUT.
3. Upon the receipt of 6th Router Solicitation from the HUT, TN1 transmits MSDU1.
4. Observe the HUT

Observable Results:

Part A

Step 2: The HUT must send a Router Solicitation with the following details

- IPv6 source address must not be set to the IPv6 unspecified address
- IPv6 destination address set to the IPv6AllRouters multicast address
- SLLA option set to the 64-bit physical address of the HUT.



Part B

Step 4: The HUT must stop retransmitting Router Solicitations. As the default routers list becomes empty, the HUT must send a Router Solicitation to find a new router at least after 30 seconds with the following details

- IPv6 source address must not be set to the IPv6 unspecified address
- IPv6 destination address set to the IPv6AllRouters multicast address
- SLLA option set to the 64-bit physical address of the HUT.

Part C

Step 4: The HUT must stop retransmitting Router Solicitations. The HUT must send a Router Solicitation to find a new prefix at least after 30 seconds, as the global prefix expires.

Part D

Step 2: The HUT must transmit multiple Router Solicitations.

- The first MAX_RTR_SOLICITATIONS (3) Router Solicitation should be separated by at least RTR_SOLICITATIONS_INTERVAL (10 seconds).
- Then HUT should perform binary exponential backoff of the retransmission timer for each subsequent transmission, but may limit the retransmission interval to MAX_RTR_SOLICITATION_INTERVAL (60 seconds).
 - The 4th Router Solicitation should be transmitted after at least 20 seconds
 - The 5th Router Solicitation should be transmitted after at least 40 seconds
 - The 6th Router Solicitation should be transmitted after at least 60 seconds

Step 4: The HUT must stop transmitting Router Solicitations.

Possible Problems:

- None.



Test 6LoWPAN.2.1.3: Processing a Router Advertisement

Purpose: Verify that the HUT properly processes a Router Advertisement and its prefix option.

References:

- [RFC 6775] –[5.4](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the all-node multicast address
MSDU2	A valid Router Advertisement with lifetime 30 seconds. It also contains a Prefix Option with the on-link flag unset and the indicated prefix as PRFX1.
MSDU3	A valid Router Advertisement with lifetime 30 seconds. It also contains a Prefix Option with the on-link flag set and the indicated prefix as PRFX1.

Table 39: Test Frames

Procedure:

Part A: Prefix Information Option with the on-link flag unset

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1 to the global address of the HUT
4. Observe the HUT.

Part B: Prefix Information Option with the on-link flag set

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU3.
3. TN1 transmits MSDU1 to the global address of the HUT
4. Observe the HUT.

Observable Results:

Part A

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 4: The HUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.2.1.4: Address configuration

Purpose: Verify that the HUT autoconfigures a global address and registers it.

References:

- [RFC 6775] – [5.4.1](#)
- [\[RFC 4862\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the all-node multicast address
MSDU2	A valid Router Advertisement with lifetime 30 seconds and the managed address configuration flag unset. It also contains a Prefix Option with the indicated prefix as PRFX1.

Table 40: Test Frames

Procedure:

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. Observe the HUT.

Observable Results:

Step 3: The HUT must transmit a unicast Neighbor Solicitation to TN1 with the following details:

- Source address is set to the global address of the HUT
- SLLA option set to the physical address of the node
- Address Registration option containing the EUI-64 address of the HUT.

Possible Problems:

- None.



Test 6LoWPAN.2.1.5: Maintaining prefix and context information

Purpose: Verify that the HUT properly handles the expiration and the update of the prefix, the context information and the default router.

References:

- [RFC 6775] –[5.4.3](#)
- [\[RFC 6282\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the HUT's global IP address from an off-link IP address.
MSDU2	A valid Router Advertisement. It also contains a Prefix Option with the on-link flag unset, a valid lifetime of 30 seconds, and the indicated prefix as PRFX1.
MSDU3	A valid Router Advertisement with lifetime 60 seconds. It also contains a Prefix Option with the on-link flag set and the indicated prefix as PRFX1. It will also contain a Context Option indicating context id 1, indicating prefix as PRFX1, a valid life time of 60 seconds, and with C Flag unset.
MSDU4	A valid Router Advertisement with lifetime 60 seconds. It also contains a Prefix Option with the on-link flag set and the indicated prefix as PRFX1. It will also contain a Context Option indicating context id 1, indicating prefix as PRFX1, a valid life time of 60 seconds, and with C Flag set.

Table 41: Test Frames

Procedure:

Part A: Expiration of the default router

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1.
4. Observe the HUT.
5. Wait for 30 seconds
6. TN1 transmits MSDU1.
7. Observe the HUT

Part B: Expiration of the prefix

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1.
4. Observe the HUT.
5. Wait for 30 seconds.
6. TN1 transmits MSDU1.
7. Observe the HUT

Part C: Expiration of the context information



1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU3.
3. TN1 transmits MSDU1.
4. Observe the HUT.
5. Wait for 60 seconds
6. TN1 transmits MSDU1.
7. Observe the HUT

Part D: Context information with compression disabled.

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU3
3. TN1 transmits MSDU4.
4. Observe the HUT.
5. TN1 transmits MSDU1.
6. Observe the HUT.

Part E: Switch to multicast Router Solicitations

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. Observe the HUT.

Observable Results:

Part A

Step 4: The HUT must transmit an ICMPv6 Echo Reply in response to an ICMPv6 Echo Request.

Step 5: Before the expiration of the default router, the HUT transmits one or more unicast Router Solicitations to TN1 in order to renew the default router information.

Step 7: The HUT must not transmit any ICMPv6 Echo Reply.

Part B

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Step 5: Before the expiration of the default router, the HUT transmits one or more unicast Router Solicitations to TN1 in order to renew the prefix information.

Step 7: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header is compressed using the context id 1.

Step 5: Before the expiration of the default router, the HUT transmits one or more unicast Router Solicitations to TN1 in order to renew the context information.

Step 7: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 4: The HUT transmits one or more a unicast Router Solicitations to TN1 in order to renew



the context information.

Step 6: The HUT should respond to MSDU with an ICMPv6 Echo Reply. The source address in the IPv6 header must not be compressed using the context id 1.

Part E

Step 3: Before the expiration of the default router, the HUT transmits one or more a unicast Router Solicitations to TN1 in order to renew the default router information. The HUT should switch back to multicast Router Solicitation.

Possible Problems:

- None.



Test 6LoWPAN.2.1.6: Registration and Neighbor Unreachability Detection

Purpose: Verify that the HUT properly performs the registration of its global addresses and handles the failure cases.

References:

- [RFC 6775] – [5.5.2](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the HUT's global IP address.
MSDU2	A valid Router Advertisement. It also contains a Prefix Option with a valid lifetime of 60 seconds and the indicated prefix as PRFX1.
MSDU3	A valid Neighbor Advertisement with status code 0. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with a lifetime of 60 seconds.
MSDU4	A valid Neighbor Advertisement with status code 1. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with a lifetime of 60 seconds.
MSDU5	A valid Neighbor Advertisement with status code 3. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with a lifetime of 60 seconds.

Table 42: Test Frames

Procedure:

Part A: Multiple Routers

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. Observe the HUT.

Part B: Successful Registration

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2
3. Upon the receipt of a Neighbor Solicitation with an ARO option, TN1 transmits MSDU3
4. TN1 transmits MSDU1
5. Observe the HUT.

Part C: Duplicate Address

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2
3. Upon the receipt of the Neighbor Solicitation with an ARO option, TN1 transmits MSDU3
4. TN1 transmits MSDU1.
5. Observe the HUT.



Part D: Duplicate Address with multiple routers

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2 twice as though there were two routers.
3. Upon the receipt of the Neighbor Solicitation with an ARO option, TN1 transmits MSDU3 as router 1 and MSDU4 as router 2.
4. Observe the HUT
5. TN1 transmits MSDU1
6. Observe the HUT

Part E: Neighbor Cache Full

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2
3. Upon the receipt of the Neighbor Solicitation with an ARO option, TN1 transmits MSDU5.
4. TN1 transmits MSDU1.
5. Observe the HUT.

Part F: Neighbor Cache Full with multiple routers

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2 twice as though there were two routers.
3. Upon the receipt of the Neighbor Solicitation with an ARO option, TN1 transmits MSDU5 as router 1 and MSDU3 as router 2.
4. TN1 transmits MSDU1.
5. Observe the HUT.

Part G: Manual DeRegistration

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. Observe the HUT.
4. Order the HUT to shut down.

Observable Results:

Part A

Step 3: The HUT should attempt to register with both routers: it should send one unicast Neighbor Solicitation to TN1:

- Source address is set to the global address of the HUT
- SLLA option set to the physical address of the node with an Address Registration option containing the EUI-64 address of the HUT

Part B

Step 5: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 5: The HUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part D

Step 4: The HUT must unregister its address with TN1 by sending a unicast Neighbor Solicitation with the following details:

- Source address is set to the global address of the HUT
- SLLA option set to the physical address of the node



- Address Registration option containing the EUI-64 address of the HUT with the lifetime set to zero.

Step 6: The HUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part E

Step 5: The HUT should not respond to MSDU1 with an ICMPv6 Echo Reply. The HUT should start transmitting multicast Router Solicitations in order to find another router.

Part F

Step 5: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The Link-layer destination address should be set to the address of Router 2.

Part G

Step 4: The HUT must unregister its address with TN1 by sending a unicast Neighbor Solicitation with the following details:

- Source address is set to the global address of the HUT
- SLLA option set to the physical address of the node
- Address Registration option containing the EUI-64 address of the HUT with the lifetime set to zero

Possible Problems:

- Part G requires that the HUT have a soft shut down mechanism.



Test 6LoWPAN.2.1.7: Recovering from Failures

Purpose: Verify that the HUT behaves correctly in the presence of a failure.

References:

- [RFC 6775] –[5.5.3](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request.
MSDU2	A valid Router Advertisement. It also contains a Prefix Option with a valid lifetime of 60 seconds and the indicated prefix as PRFX1.
MSDU3	A valid Neighbor Advertisement. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with status 0.
MSDU4	A valid Neighbor Advertisement. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with status 80.
MSDU5	A valid Neighbor Advertisement. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with status 1.
MSDU6	A valid Neighbor Advertisement. It also contains a Target Address Option with the HUT's link-local address. It also contains an Address Registration Option with status 2.

Table 43: Test Frames

Procedure:

Part A: NUD Timeout

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2 twice as though there were two routers.
3. Upon the receipt of the Neighbor Solicitation from the HUT, Wait 90 seconds.
4. TN1 transmits MSDU1 to the HUT.
5. Observe the HUT.

Part B: Status indicating an error.

1. Connect HUT to TN1.
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU4.
3. TN1 transmits MSDU1 to the HUT.
4. Observe the HUT.

Part C: Other routers de-registration

1. Initialize the network interface of the HUT
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 acts as two routers, sending MSDU3 as router 1 and MSDU5 as Router 2.
3. TN1 transmits MSDU1 to the HUT.
4. Observe the HUT.

Part D: Status indicating a full Router's neighbor cache



1. Connect HUT to TN1.
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU6.
3. TN1 transmits MSDU1 to the HUT.
4. Observe the HUT.

Observable Results:

Part A

Step 5: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Part B

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 Echo Reply.

Part C

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 echo reply. The HUT must send a Router Solicitation to TN1 with ARO lifetime set to zero.

Part D

Step 4: The HUT should not reply to MSDU1 with an ICMPv6 echo reply. The HUT must send a Router Solicitation to the IPv6AllRouters multicast address.

Possible Problems:

- None.



Test 6LoWPAN.2.1.8: Next-hop Determination

Purpose: Verify that the HUT properly chooses the next hop according to the type and scope of the destination address.

References:

- [RFC 6775] – [5.6](#)
- [\[RFC 4944\]](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request.
MSDU2	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the HUT's global address.
MSDU3	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the all-node multicast address.

Table 44: Test Frames

Procedure:

Part A: Link-local address

1. Connect HUT to TN1.
2. TN1 transmits MSDU1 to the HUT.
3. Observe the HUT.

Part B: Global address off-link

1. Configure TN1 to use prefix PRFX2::1.
2. TN1 transmits MSDU2.
3. Observe the HUT.

Part C: Global address on-link

1. Connect HUT to TN1.
2. Configure TN1 with a global address.
3. TN1 transmits MSDU2.
4. Observe the HUT.

Part D: Multicast Address on-link

1. Connect HUT to TN1.
2. Configure TN1 with a global address.
3. TN1 transmits MSDU3
4. Observe the HUT.

Observable Results:

Part A

Step 3: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The Link-layer destination address must be set to the address of TN1.



Part B

Step 3: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The Link-layer destination address must be set to the address of TN1.

Part C

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The Link-layer destination address must be set to the address of TN1.

Part D

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The Link-layer destination address must be set to the address of TN1.

Possible Problems:

- None



Test 6LoWPAN.2.1.9: Behavior on wake-up

Purpose: Verify that the HUT properly handles registration that allows sleeping mode.

References:

- [RFC 6775] – [5.8.1](#)
- [\[RFC 4944\]](#)
- [\[RFC 4861\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid Neighbor Advertisement with status 0. It also has a target link-local address of the HUT and an Address Registration Option with lifetime of 60 seconds.

Table 45: Test Frames

Procedure:

Part A: Maintaining address registration on sleeping

1. Connect HUT and TN1.
2. Upon the receipt of the Neighbor Solicitation from the HUT, TN1 transmits MSDU1 to the HUT.
3. Ask the HUT to sleep for 60 seconds
4. Observe the HUT.

Part B: Maintaining default routers on wake-up

1. Initialize the network interface of the HUT.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU1 to the HUT.
3. Ask the HUT to sleep for 60 seconds
4. Observe the HUT.

Observable Results:

Part A

Step 4: Once woke up, The HUT should send a Neighbor Solicitation with an address registration option containing the EUI-64 address of the HUT with the lifetime set to 60 to TN1 to maintain its registration.

Part B

Step 8: Once woke up, the HUT must send a Router Solicitation to the IPv6AllRouters multicast address.

Possible Problems:

- The HUT to sleep must have the ability to manually be put asleep for this test case.



Test 6LoWPAN.2.1.10: Storing Contexts

Purpose: Verify that the HUT properly handle context used for compression.

References:

- [RFC 6775] – [5.4.2](#)
- [\[RFC 4944\]](#)
- [\[RFC 4861\]](#)
- [\[RFC 6282\]](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request.
MSDU2	A valid Router Advertisement with lifetime 60 seconds. It also has a prefix option with prefix PRFX1 and valid lifetime 60 seconds. It also has a context option with context id 1, c flag unset, a lifetime of 60 seconds, and prefix PRFX1.
MSDU3	A valid Router Advertisement with lifetime 60 seconds. It also has a prefix option with prefix PRFX1 and valid lifetime 60 seconds. It also has a context option with context id 1, c flag unset, a lifetime of 120 seconds, and prefix PRFX1.
MSDU4	A valid Router Advertisement with lifetime 60 seconds. It also has a prefix option with prefix PRFX1 and valid lifetime 60 seconds. It also has a context option with context id 1, c flag unset, a lifetime of 0 seconds, and prefix PRFX1.
MSDU5	A valid Router Advertisement with lifetime 60 seconds. It also has a prefix option with prefix PRFX1 and valid lifetime 60 seconds. It also has a context option with context id 2, c flag unset, a lifetime of 60 seconds, and prefix PRFX1.

Table 46: Test Frames

Procedure:

Part A: Expiration of the context information

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1 to HUT's global address from an off-line address.
4. Observe the HUT.
5. Wait for 60 seconds
6. TN1 transmits MSDU1 to HUT's global address from an off-line address.
7. Observe the HUT.

Part B: Updating the context information.

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1 to HUT's global address from an off-line address.
4. Observe the HUT.
5. TN1 transmits MSDU3 to the HUT.
6. Observe the HUT.



7. Wait for 60 seconds.
8. TN1 transmits MSDU1 to HUT's global address from an off-line address.
9. Observe the HUT.

Part C: Receiving a the context information with a zero lifetime

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU2.
3. TN1 transmits MSDU1 to HUT's global address from an off-line address.
4. Observe the HUT.
5. Wait for 30 seconds.
6. TN1 transmits MSDU4 to HUT.
7. TN1 transmits MSDU1 to HUT's global address from an off-line address.
8. Observe the HUT.

Part D: adding a new the context information.

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT.
3. TN1 transmits MSDU2.
4. TN1 transmits MSDU5.
5. TN1 transmits MSDU1 to HUT's global address from an off-line address.
6. Observe the HUT.
7. TN1 transmits MSDU1 to HUT's global address from an off-line address.
8. Observe the HUT.

Observable Results:

Part A

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Step 9: The HUT must send any Echo Reply. The source address in the IPv6 header must not be compressed using the context id 1.

Part B

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Step 6: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Step 9: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Part C

Step 4: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Step 8: The HUT should not respond to MSDU1 with an ICMPv6 Echo Reply.

Part D



Step 6: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 1.

Step 8: The HUT should respond to MSDU1 with an ICMPv6 Echo Reply. The source address in the IPv6 header should be compressed using the context id 2.

Possible Problems:

- None.



Test 6LoWPAN.2.1.11: Address Registration Option Formatting

Purpose: Verify that the HUT properly includes the new address registration option (ARO) when sending Neighbor solicitation to a router.

References:

- [RFC 6775] – [4.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid Router Solicit Message with prefix option for prefix PRFX1 with the M bit set to 0.

Table 47: Test Frames

Procedure:

1. Connect HUT to TN1.
2. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU1.
3. Observe the HUT.

Observable Results:

Step 3: The HUT must send a unicast Neighbor Solicitation to TN1 with the following details:

- An Address Registration option with:
 - Type: 33
 - Length: 2
 - Status: 0
 - Reserved: 0
 - EUI-64: EUI-64 address of the HUT

Possible Problems:

- None.



Subsection 2: Router Behavior

Scope

The following tests cover the behavior of a router implementing the Neighbor Discovery optimizations for 6LoWPAN.

Overview

Test in this group verify that a router properly handle 6LoWPAN specific behavior, packets format, and options described in [RFC 6775] section [4](#) and [6](#). This includes particularly:

- The handling of Context Options.
- The handling of Authoritative Border Router Options.
- The host-initiated interactions for Router Advertisement/Solicitation messages.
- The host-initiated interactions for Neighbor Advertisement/Solicitation.
- The optional multi-hop duplicate address detection.
- The support of a sleeping mode for nodes.



Test 6LoWPAN.2.2.1: Forbidden action

Purpose: Verify that a RUT doesn't set the (on-link) L bit when sending a router advertisement.

References:

- [RFC 6775] – [6.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Router Solicit Message with source link-layer option set to the link-local address of TN1.

Table 48: Test Frames

Procedure:

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. Observe the RUT.

Observable Results:

Step 3: The RUT should respond to MSDU1 with a Router Advertisement with the following details:

- The on-link (L) bit unset (0).

Possible Problems:

- None.



Test 6LoWPAN.2.2.2: Processing a Router Solicitation

Purpose: Verify that the RUT properly handles and answers to the Router Solicitations (RS) sent by nodes.

References:

- [RFC 6775] – [6.3](#)
- [RFC 4861] – [7.3.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Router Solicit Message with source link-layer option set to the link-local address of TN1.
MSDU2	A valid ICMPv6 Echo Request message.

Table 49: Test Frames

Procedure:

Part A: Sending router advertisement via unicast.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1
3. Observe the RUT.

Part B: Modification of the cache entries.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2 to the RUT
3. Observe the RUT.

Part C: Inclusion of the source link-layer address option.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1
3. Observe the RUT.

Observable Results:

Part A

Step 2: The RUT should respond to MSDU1 with a Router Advertisement message with the destination address set to unicast address of TN1.

Part B

Step 4: The RUT should not respond to MSDU2 with an ICMPv6 Echo Reply. Instead, The RUT should transmit a Neighbor Solicitation to TN1.

Part C

Step 6: The RUT should respond to MSDU1 with a Router Advertisement with a Source Link-Layer Address (SLLA) Option set to the 64-bit physical address of the RUT.



Possible Problems:

- None.



Test 6LoWPAN.2.2.3: Processing a Neighbor Solicitation

Purpose: Verify the RUT can properly handle and answer Neighbor Solicitations sent by other nodes.

References:

- [RFC 6775] – [6.5](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with a lifetime of 60 seconds, the EUI-64 address of TN1, a length of 9, and a status of 0.
MSDU2	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with a lifetime of 60 seconds, the EUI-64 address of TN1, a length of 80, and a status of 0.
MSDU3	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with a lifetime of 60 seconds, the EUI-64 address of the unspecified address, a length of 2, and a status of 0.
MSDU4	A valid Neighbor Solicitation Message. It has an address registration option with a lifetime of 60 seconds, the EUI-64 address of TN1, a length of 2, and a status of 0.

Table 50: Test Frames

Procedure:

Part A: Receiving a Neighbor Solicitation with wrong length

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. Observe the RUT.

Part B: Receiving a Neighbor Solicitation with wrong status

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2.
3. Observe the RUT.

Part C: Receiving a Neighbor Solicitation with ARO from an unspecified address

1. Connect TN1 to RUT.
2. TN1 transmits MSDU3.
3. Observe the RUT.

Part D: Receiving a Neighbor Solicitation with an ARO without a SLLA option

1. Connect TN1 to RUT.
2. TN1 transmits MSDU4.
3. Observe the RUT.



Observable Results:

Part A

Step 3: The RUT should silently discard MSDU1. It should not send any Neighbor Advertisement in response to the MSDU1.

Part B

Step 3: The RUT should silently discard MSDU2. It should not send any Neighbor Advertisement in response to MSDU2.

Part C

Step 3: The RUT should respond to MSDU3 with a Neighbor Advertisement as if it received a Neighbor Solicitation without an Address Registration Option. The RUT should not send any Duplicate Address request.

Part D

Step 3: The RUT should respond to MSDU3 with a Neighbor Advertisement as if it received a Neighbor Solicitation without an Address Registration Option. The RUT should not send any Duplicate Address request.

Possible Problems:

- None.



Test 6LoWPAN.2.2.4: Duplicate Address Detection Behavior

Purpose: Verify that the RUT properly uses the duplicate address detection (single-hop and multi-hop) mechanism.

References:

- [RFC 6775] – [6.5.1](#)
- [RFC 6775] – [8.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with the EUI-64 address of TN1.
MSDU2	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address with prefix PRFX1:42::1. It also has an address registration option with the EUI-64 address 12-34-56-78-90-00-00-00.
MSDU3	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address with prefix PRFX1:42::1. It also has an address registration option with the EUI-64 address 12-34-56-78-90-00-00-11.

Table 51: Test Frames

Procedure:

Part A: Receiving a Neighbor Solicitation with an ARO for an address not in cache.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. Observe the RUT.

Part B: Receiving a Neighbor Solicitation with an ARO for an address in cache, with the same EUI-64.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2.
3. Observe the RUT.

Part C: Receiving a Neighbor Solicitation with an ARO for an address in cache, with a different EUI-64

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2.
3. TN2 transmits MSDU3.
4. Observe the RUT.

Part D: Using the Destination Address given by the SSLA option.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2.
3. TN2 transmits MSDU3.
4. Observe the RUT.



Observable Results:

Part A

Step 3: The RUT should respond to MSDU1 with a Neighbor Advertisement with an Address Registration Option (ARO) with the status field set to zero (0).

Part B

Step 3: The RUT should respond to MSDU2 with a Neighbor Advertisement with an Address Registration Option (ARO) with the status field set to zero (0).

Part C

Step 4: The RUT should respond to MSDU3 with a Neighbor Advertisement with an Address Registration Option (ARO) with the status field set to zero (1).

Part D

Step 4: The RUT should respond to MSDU3 with a Neighbor Advertisement with an IPv6 destination address set to the address set to value given in the SLLA option in the Neighbor Solicitation.

Possible Problems:

- None.



Test 6LoWPAN.2.2.5: Returning Address Registration Errors

Purpose: Verify that the RUT can properly choose the address to send Neighbor Advertisement when a duplicate address has been detected.

References:

- [RFC 6775] – [6.5.1](#)
- [RFC 6775] – [6.5.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address with prefix PRFX1:42::1. It also has an address registration option with the EUI-64 address 12-34-56-78-90-00-00-00.
MSDU2	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address with prefix PRFX1:42::1. It also has an address registration option with the EUI-64 address 12-34-56-78-90-00-00-11.
MSDU3	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address with prefix PRFX1:42::1. It also has an address registration option with the EUI-64 address 12-34-56-78-90-00-00-00 and lifetime of 2280 seconds.

Table 52: Test Frames

Procedure:

Part A: Using the IPv6 Destination address derived from ARO.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1
3. TN1 transmits MSDU2
4. Observe the RUT.

Part B: Including a copy of the ARO.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU3.
3. TN1 transmits MSDU3.
4. Observe the RUT.

Observable Results:

Part A

Step 3: The RUT should respond to MSDU1 with a Neighbor Advertisement to TN1. The Neighbor Advertisement should not use the original IPv6 source address that is already in use. The Neighbor Advertisement should have a destination address derived from EUI-64 field in ARO.

Part B



Step 4: The RUT should respond to MSDU3 with a Neighbor Advertisement to TN1 that should include a copy of the ARO received with the EUI-64 set to 12-34-56-78-90-00-00-11 and a lifetime field set to 38 (2280 seconds).

Possible Problems:

- None.



Test 6LoWPAN.2.2.6: Updating the Neighbor Cache

Purpose: Verify that the RUT correctly updates the neighbor cache when it receives Neighbor Solicitations.

References:

- [RFC 6775] – [6.5.3](#)
- [RFC 6775] – [6.6](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with the EUI-64 address of TN1 and a lifetime of 0.

Table 53: Test Frames

Procedure:

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1
3. Observe the RUT.

Observable Results:

Step 3: The RUT must delete the cache entry and transmit a Neighbor Advertisement (NA) with the Address Registration Option set to zero (0) in response to MSDU1.

Possible Problems:

- None.



Test 6LoWPAN.2.2.7: Next-Hop Determination

Purpose: Verify that the RUT can properly choose the next hop.

References:

- [RFC 6775] –[6.5.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with the EUI-64 address of TN1 and a lifetime of 60.
MSDU2	A valid IPv6 frame containing an ICMPv6 Echo Request.

Table 54: Test Frames

Procedure:

Part A: Next Hop determination for an address present in cache.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. TN1 transmits MSDU2 to RUT.
4. Observe the RUT.

Part B: Next Hop determination for an address not present in cache.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU2 to RUT.
3. Observe the RUT.

Part C: Disallow of Tentative Neighbor cache entries for maintaining on link status of a node.

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. Wait for 60 seconds.
4. TN1 transmits MSDU2 to RUT.
5. Observe the RUT.

Observable Results:

Part A

Step 3: The RUT should respond to MSDU2 with an ICMPv6 echo reply with to TN1's link layer address with a link-local destination address of TN1 link local address.

Part B

Step 3: The RUT must transmit an ICMPv6 Echo Request to TN1's link layer address with a destination address of TN1 link local address.

Part C



Step 5: The RUT must transmit an ICMPv6 Echo Request to TN1's link layer address with a destination address of TN1 link local address.

Possible Problems:

- None.



Test 6LoWPAN.2.2.8: Address Resolution Between Routers

Purpose: Verify that the RUT can properly resolve addresses between other routers.

References:

- [RFC 6775] – [6.5.5](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with the Pure Topology in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TR1. It also has an address registration option with the EUI-64 address of TR1 and a lifetime of 60. The message should be directed to the solicited node multicast address of the RUT.
MSDU2	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the solicited multicast address of the RUT.

Table 55: Test Frames

Procedure:

Part A: Joining the solicited node multicast address

1. Connect TR1 to RUT.
2. TR1 transmits MSDU1.
3. Observe the RUT.

Part B: Joining the solicited node multicast address

1. Connect TR1 to RUT.
2. TR1 transmits MSDU2.
3. Observe the RUT.

Observable Results:

Part A

Step 2: The RUT should respond the MSDU1 with a Neighbor Advertisement.

Part B

Step 4: The RUT should respond to MSDU2 with an ICMPv6 Echo Reply.

Possible Problems:

- None



Test 6LoWPAN.2.2.9: Router-to-Router Interaction

Purpose: To verify that the RUT can correctly communicate with another 6LRouter.

References:

- [RFC 6775] – [3.4](#)
- [RFC 6775] – [4.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT Pure Topology configuration in this test case.

Test Frames	
MSDU1	A valid IPv6 frame containing an ICMPv6 Echo Request directed to the RUT
MSDU2	A valid ICMPv6 Router Advertise message.
MSDU3	A valid ICMPv6 Router Solicit message.
MSDU4	A Duplicate Address Confirmation message.

Table 56: Test Frames

Procedure:

Part A: Initial Startup with EUID-64 Addresses

1. Configure TR1 and the RUT to use EUID-64 link-local Address.
2. Connect the RUT to TR1.
3. Observe RUT.
4. TR1 transmits MSDU2 to RUT.
5. Observe RUT.
6. TR1 transmits MSDU3 to RUT.
7. Observe RUT.
8. TR1 transmits MSDU1 to the RUT.
9. Observe RUT.

Part B: Initial Startup with 16-bit Short Addresses

1. Configure the PAN coordinator to issue 16-bit short addresses to the RUT and TR1.
2. Connect the RUT to TR1.
3. Observe RUT.
4. TR1 transmits MSDU2 to RUT.
5. Observe RUT.
6. TR1 transmits MSDU3 to RUT.
7. Observe RUT.
8. TR1 transmits MSDU1 to the RUT.
9. Observe RUT.

Part C: Multihop Prefix Used

1. Configure TR1 and the RUT to use EUID-64 link-local Address.
2. Configure TR1 and the RUT to use the multihop prefix.
3. Connect the RUT to TR1.
4. Observe RUT.
5. TR1 transmits MSDU2 to RUT.



6. Observe RUT.
7. TR1 transmits MSDU3 to RUT.
8. Observe RUT.
9. TR1 transmits MSDU1 to the RUT.
10. Observe RUT.

Part D: Context Dissemination Used

1. Configure TR1 and the RUT to use EUID-64 link-local Address.
2. Configure TR1 and the RUT to disseminate contexts.
3. Connect the RUT to TR1.
4. Observe RUT.
5. TR1 transmits MSDU2 to RUT.
6. Observe RUT.
7. TR1 transmits MSDU3 to RUT.
8. Observe RUT.
9. TR1 transmits MSDU1 to the RUT.
10. Observe RUT.

Part E: Duplicate Address Detection

1. Configure TR1 and the RUT to use EUID-64 link-local Address.
2. Configure TR1 and the RUT with a default hop limit of 5.
3. Connect the RUT to TR1.
4. Observe RUT.
5. TR1 transmits MSDU4 to RUT.
6. Observe RUT.
7. TR1 transmits MSDU1 to the RUT.
8. Observe RUT.
9. Repeat steps 2-8 for default hop limit of 255.

Observable Results:

Part A

Step 3: The RUT should send a Router Solicit message to the all-router multicast address.

Step 5: The RUT should receive MSDU2 without issue.

Step 7: The RUT should respond to MSDU3 with a valid Router Advertise message.

Step 9: The RUT should respond to MSUD1 with an ICMPv6 Echo Reply.

Part B

Step 3: The RUT should send a Router Solicit message to the all-router multicast address.

Step 5: The RUT should receive MSDU2 without issue.

Step 7: The RUT should respond to MSDU3 with a valid Router Advertise message.

Step 9: The RUT should respond to MSUD1 with an ICMPv6 Echo Reply.

Part C

Step 4: The RUT should send a Router Solicit message to the all-router multicast address.



Step 6: The RUT should receive MSDU2 without issue.

Step 8: The RUT should respond to MSDU3 with a valid Router Advertise message. The Router Advertisement should contain the ABRO, 6CO, and prefix information from TR1.

Step 10: The RUT should respond to MSUD1 with an ICMPv6 Echo Reply.

Part D

Step 4: The RUT should send a Router Solicit message to the all-router multicast address.

Step 6: The RUT should receive MSDU2 without issue.

Step 8: The RUT should respond to MSDU3 with a valid Router Advertise message. The Router Advertisement should contain the ABRO, 6CO, and prefix information from TR1.

Step 10: The RUT should respond to MSUD1 with an ICMPv6 Echo Reply.

Part E

Step 4: The RUT should send a Duplicate Address Request message.

Step 6: The RUT should receive MSDU4 without issue.

Step 8: The RUT should respond to MSUD1 with an ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.2.2.10: Address Registration Option Format

Purpose: Verify that the RUT properly includes the new address registration option (ARO) when sending Neighbor Advertisement (NA) to a node.

References:

- [RFC 6775] – [4.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Neighbor Solicitation Message with source link-layer option set to the link-local address of TN1. It also has an address registration option with the EUI-64 address of TN1 and a lifetime of 60.

Table 57: Test Frames

Procedure:

1. TN1 transmits MSDU1 to the RUT.
2. Observe the RUT.

Observable Results:

Step 2: The RUT should respond to MSDU1 with a unicast Neighbor Advertisement with an Address Registration Option containing the following:

- Type: 33
- Length: 2
- Status: 0
- Reserved: 0
- EUI-64: EUI-64 address of TN1

Possible Problems:

- None.



Test 6LoWPAN.2.2.11: Context Option Format

Purpose: Verify that the RUT properly includes the new context option (6CO) when sending Router Advertisements (RA) to a node or a router.

References:

- [RFC 6775] – [4.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Router Solicitation Message with source link-layer option set to the link-local address of TN1.

Table 58: Test Frames

Procedure:

1. Connect TN1 to RUT.
2. TN1 transmits a MSDU1.
3. Observe the RUT.

Observable Results:

Step 3: The RUT should respond to MSDU1 with a unicast Router Advertisement (RA) to TN1 with the following:

- A Context Option (6CO) with:
 - Type: 34
 - Length: 2
 - Context-length: 64
 - Reserved: 0
 - (C) flag: 1
 - Context id: 1
 - Reserved: 0
 - Lifetime: 60
 - Context Prefix: PRFX1

Possible Problems:

- None.



Test 6LoWPAN.2.2.12: Authoritative Border Router Option Format

Purpose: Verify that the RUT properly includes the new Authoritative Border Router Option (ABRO) when sending Router Advertisements (RA) to a node or to a router.

References:

- [RFC 6775] – [4.3](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Router Solicitation Message with source link-layer option set to the link-local address of TN1.

Table 59: Test Frames

Procedure:

1. Connect TN1 to RUT.
2. TN1 transmits MSDU1.
3. Observe the RUT.

Observable Results:

Step 3: The RUT must send a unicast Router Advertisement to TN1 containing the following Authoritative Border Router Option:

- Type: 35
- Length: 3
- Version: 1
- Reserved: 0
- Address: TN1's address

Possible Problems:

- None.



Section 6: RFC 7400

Scope

The following tests cover the base specification for Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks, Request For Comments 7400.

Overview

The tests in this group verify that a node properly handle GHC as an alternative NHC for IPHC.



Group 1: Frame Format

Scope

The following tests cover the Dispatch Header, interface identifier generation, and reserved header fields.

Overview

The tests in this group verify the dispatch type and headers in LoWPAN Adaption layer.



Test 6LoWPAN.1.1.1: Host 6LoWPAN Capabilities Indication Option Formatting

Purpose: Verify that the HUT properly handles the 6LoWPAN Capabilities Indication Option.

References:

- [RFC 7400] – [3.3](#)
- [RFC 7400] – [3.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPHC IPv6 frame containing an uncompressed ICMPv6 Echo Request.
MSDU2	A valid IPHC IPv6 frame containing a GHC ICMPv6 Echo Request.
MSDU3	A valid Router Advertisement with a 6CIO with the G bit set.
MSDU4	A valid Router Advertisement with a 6CIO with the G bit unset.
MSDU5	A valid Router Advertisement with a 6CIO with the G bit set and the reserved fields set.
MSDU6	A valid Router Advertisement with a 6CIO with the G bit set and a length of 2.

Table 60: Test Frames

Procedure:

Part A: Reception of a valid 6CIO with G bit set

1. Enable GHC on both the HUT and TN1.
2. Connect HUT to TN1.
3. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU3.
4. TN1 transmits MSDU2 to HUT.
5. Observe the HUT.

Part B: Reception of a valid 6CIO with G bit not set

1. Enable GHC on the HUT.
2. Connect HUT to TN1.
3. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU4.
4. TN1 transmits MSDU1 to HUT.
5. Observe the HUT.

Part C: Reception of a valid 6CIO with G bit unset but GHC is supported

1. Enable GHC on both the HUT and TN1.
2. Connect HUT to TN1.
3. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU4.
4. TN1 transmits MSDU2 to HUT.
5. Observe the HUT.

Part D: Reception of a 6CIO with reserved bits set

1. Enable GHC on both the HUT and TN1.
2. Connect HUT to TN1.
3. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU5.
4. TN1 transmits MSDU2 to HUT.
5. Observe the HUT.



Part E: Reception of a 6CIO with invalid length field

1. Enable GHC on both the HUT and TN1.
2. Connect HUT to TN1.
3. Upon the receipt of the Router Solicitation from the HUT, TN1 transmits MSDU6.
4. Observe the HUT.
5. TN1 transmits MSDU2 to HUT.
6. Observe the HUT.

Observable Results:

Part A

Step 3: The HUT should include a 6CIO in its router solicitation message with the G bit set.

Step 5: The HUT should reply to MSDU2 with an ICMPv6 Echo Reply that is compressed using GHC.

Part B

Step 3: The HUT should include a 6CIO in its router solicitation message with the G bit set.

Step 5: The HUT should reply to MSDU1 with an uncompressed ICMPv6 Echo Reply.

Part C

Step 3: The HUT should include a 6CIO in its router solicitation message with the G bit set.

Step 5: The HUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part D

Step 3: The HUT should include a 6CIO in its router solicitation message with the G bit set.

Step 5: The HUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part E

Step 3: The HUT should include a 6CIO in its router solicitation message with the G bit set.

Step 4: The HUT should silently discard MSDU6.

Step 6: The HUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.2: Router 6LoWPAN Capabilities Indication Option Formatting

Purpose: Verify that the RUT properly handles the 6LoWPAN Capabilities Indication Option.

References:

- [RFC 7400] – [3.3](#)
- [RFC 7400] – [3.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT using the Hybrid Topology in this test case.

Test Frames	
MSDU1	A valid IPHC IPv6 frame containing an uncompressed ICMPv6 Echo Request.
MSDU2	A valid IPHC IPv6 frame containing a GHC ICMPv6 Echo Request.
MSDU3	A valid Router Solicit with a 6CIO with the G bit set.
MSDU4	A valid Router Solicit with a 6CIO with the G bit unset.
MSDU5	A valid Router Solicit with a 6CIO with the G bit set and the reserved fields set.
MSDU6	A valid Router Solicit with a 6CIO with the G bit set and a length of 2.

Table 61: Test Frames

Procedure:

Part A: Reception of a valid 6CIO with G bit set

1. Enable GHC on both the RUT and TN1.
2. Connect RUT to TN1.
3. TN1 transmits MSDU3.
4. Observe the RUT.
5. TN1 transmits MSDU2 to RUT.
6. Observe the RUT.

Part B: Reception of a valid 6CIO with G bit not set

1. Enable GHC on the RUT.
2. Connect RUT to TN1.
3. TN1 transmits MSDU4.
4. Observe the RUT.
5. TN1 transmits MSDU1 to RUT.
6. Observe the RUT.

Part C: Reception of a valid 6CIO with G bit unset but GHC is supported

1. Enable GHC on both the RUT and TN1.
2. Connect RUT to TN1.
3. TN1 transmits MSDU4.
4. Observe the RUT.
5. TN1 transmits MSDU2 to RUT.
6. Observe the RUT.

Part D: Reception of a 6CIO with reserved bits set

1. Enable GHC on both the RUT and TN1.
2. Connect RUT to TN1.



3. TN1 transmits MSDU5.
4. Observe the RUT.
5. TN1 transmits MSDU2 to RUT.
6. Observe the RUT.

Part E: Reception of a 6CIO with invalid length field

1. Enable GHC on both the RUT and TN1.
2. Connect RUT to TN1.
3. TN1 transmits MSDU6.
4. Observe the RUT.
5. TN1 transmits MSDU2 to RUT.
6. Observe the RUT.

Observable Results:

Part A

Step 4: The RUT should include a 6CIO in its router advertisement message with the G bit set.

Step 6: The RUT should reply to MSDU2 with an ICMPv6 Echo Reply that is compressed using GHC.

Part B

Step 4: The RUT should include a 6CIO in its router advertisement message with the G bit set.

Step 6: The RUT should reply to MSDU1 with an uncompressed ICMPv6 Echo Reply.

Part C

Step 4: The RUT should include a 6CIO in its router advertisement message with the G bit set.

Step 6: The RUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part D

Step 4: The RUT should include a 6CIO in its router advertisement message with the G bit set.

Step 6: The RUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part E

Step 4: The RUT should silently discard MSDU6.

Step 6: The RUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Possible Problems:

- None.



Test 6LoWPAN.1.1.3: Neighbor Discovery 6LoWPAN Capabilities Indication Option Formatting

Purpose: Verify that the NUT properly handles the 6LoWPAN Capabilities Indication Option when using neighbor discovery.

References:

- [RFC 7400] – [3.3](#)
- [RFC 7400] – [3.4](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT using both the Pure and Hybrid topologies.

Test Frames	
MSDU1	A valid IPHC IPv6 frame containing an uncompressed ICMPv6 Echo Request.
MSDU2	A valid IPHC IPv6 frame containing a GHC ICMPv6 Echo Request.
MSDU3a	A valid Neighbor Solicit with a 6CIO with the G bit set.
MSDU3b	A valid Neighbor Advertisement with a 6CIO with the G bit set.
MSDU4a	A valid Neighbor Solicit with a 6CIO with the G bit unset.
MSDU4b	A valid Neighbor Advertisement with a 6CIO with the G bit unset.
MSDU5a	A valid Neighbor Solicit with a 6CIO with the G bit set and the reserved fields set.
MSDU5b	A valid Neighbor Advertisement with a 6CIO with the G bit set and the reserved fields set.
MSDU6a	A valid Neighbor Solicit with a 6CIO with the G bit set and a length of 2.
MSDU6b	A valid Neighbor Advertisement with a 6CIO with the G bit set and a length of 2.

Table 62: Test Frames

Procedure:

Part A: Reception of a valid 6CIO with G bit set

1. Enable GHC on both the NUT and TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU3a
4. Observe the NUT.
5. Upon the receipt of the Neighbor Solicitation from the NUT, TN1 transmits MSDU3b.
6. TN1 transmits MSDU2 to NUT.
7. Observe the NUT.

Part B: Reception of a valid 6CIO with G bit not set

1. Enable GHC on both the NUT and TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU4a
4. Observe the NUT.
5. Upon the receipt of the Neighbor Solicitation from the NUT, TN1 transmits MSDU4b.
6. TN1 transmits MSDU1 to NUT.
7. Observe the NUT.

Part C: Reception of a valid 6CIO with G bit unset but GHC is supported



1. Enable GHC on both the NUT and TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU4a
4. Observe the NUT.
5. Upon the receipt of the Neighbor Solicitation from the NUT, TN1 transmits MSDU4b.
6. TN1 transmits MSDU2 to NUT.
7. Observe the NUT.

Part D: Reception of a 6CIO with reserved bits set

1. Enable GHC on both the NUT and TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU5a
4. Observe the NUT.
5. Upon the receipt of the Neighbor Solicitation from the NUT, TN1 transmits MSDU5b.
6. TN1 transmits MSDU2 to NUT.
7. Observe the NUT.

Part E: Reception of a 6CIO with invalid length field

1. Enable GHC on both the NUT and TN1.
2. Connect NUT to TN1.
3. TN1 transmits MSDU6a
4. Observe the NUT.
5. Upon the receipt of the Neighbor Solicitation from the NUT, TN1 transmits MSDU6b.
6. Observe the NUT.
7. TN1 transmits MSDU2 to NUT.
8. Observe the NUT.

Observable Results:

Part A

Step 4: The NUT should include a 6CIO in its neighbor advertise message with the G bit set.

Step 7: The NUT should reply to MSDU2 with an ICMPv6 Echo Reply that is compressed using GHC.

Part B

Step 4: The NUT should include a 6CIO in its neighbor advertise message with the G bit set.

Step 7: The NUT should reply to MSDU1 with an uncompressed ICMPv6 Echo Reply.

Part C

Step 4: The NUT should include a 6CIO in its neighbor advertise message with the G bit set.

Step 7: The NUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part D

Step 4: The NUT should include a 6CIO in its neighbor advertise message with the G bit set.

Step 7: The NUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Part E



Step 4: The NUT should silently discard MSDU6a.

Step 6: The NUT should silently discard MSDU6b.

Step 8: The NUT should reply to MSDU2 with a GHC ICMPv6 Echo Reply.

Possible Problems:

- None.



Group 2: Header Compression Operations

Scope

The following tests cover the different compression mechanisms.

Overview

The tests in this group verify that the NUT is capable of correctly compressing 6LoWPAN packets using HC1, IPHC, NHC, and GHC.

Test 6LoWPAN.2.1.1: LOWPAN_GHC IPv6 Extension Header Encoding

Purpose: To verify that the NUT can correctly compress the IPv6 extension headers using GHC.

References:

- [RFC 7400] – [3.2](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPHC compressed IPv6 packet with hop-by-hop option header.
MSDU2	A valid IPHC compressed IPv6 packet with router header.
MSDU3	A valid IPHC compressed IPv6 packet with fragmentation header.
MSDU4	A valid IPHC compressed IPv6 packet with destination option header.
MSDU5	A valid IPHC compressed IPv6 packet with mobility header.
MSDU6	A valid IPHC compressed IPv6 packet with another IPv6 Header followed by an ICMPv6 Echo Request.

Table 63: Test Frames

Procedure:

Part A: IPHC IPv6 Hop-by Hop Option Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to NUT.
5. Observe NUT.

Part B: IPHC IPv6 Routing Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU2 to NUT.
5. Observe NUT.

Part C: IPHC IPv6 Fragment Header



1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU3 to NUT.
5. Observe NUT.

Part D: IPHC IPv6 Destination Options Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU4 to NUT.
5. Observe NUT.

Part E: IPHC IPv6 Mobility Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU5 to NUT.
5. Observe NUT.

Part F: IPHC Additional IPv6 Header

1. Configure the NUT and TN1 to use EUI-64 link-local Address.
2. Enable GHC on both the NUT and TN1.
3. Connect NUT to TN1.
4. TN1 transmits MSDU6 to NUT.
5. Observe NUT.

Observable Results:

Part A

Step 5:

Part B

Step 5:

Part C

Step 5:

Part D

Step 5:

Part E

Step 5:

Part F

Step 5:

Possible Problems:

- None.



Test 6LoWPAN.2.1.2: LOWPAN_GHC Compressed UDP Encoding

Purpose: To verify that the NUT can correctly encode a UDP frame using GHC encoding.

References:

- [RFC 7400] – [3.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header containing a CoAP ping. The IPv6 header shall be compressed using IPHC. The UDP next header will also be compressed using GHC.

Table 64: Test Frames

Procedure:

Part A: LOWPAN_IPHC UDP Port compressing with EUI-64 Addressing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Enable GHC on TN1 and the NUT.
3. Configure TN1's CoAP Ping Server on port 61440.
4. Configure the NUT's CoAP Ping Server on port 61430.
5. Connect NUT to TN1.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.
8. Repeat steps 2-6 with the following source port configurations. {61440,61445, 61490,61470,61695 }
9. Configure TN1's CoAP Ping Server on port 61430.
10. Configure the NUT's CoAP Ping Server on port 61440.
11. Connect NUT to TN1.
12. TN1 transmits MSDU1 to the NUT.
13. Observe NUT.
14. Repeat steps 8-12 with the following destination port configurations. {61440,61445, 61490,61470,61695 }

Part B: LOWPAN_IPHC UDP Port compressing with 16-bit Short Addressing

1. Configure the PAN coordinator to issue 16-bit Short addresses to TN1 and the NUT.
2. Enable GHC on TN1 and the NUT.
3. Configure TN1's CoAP Ping Server on port 61440.
4. Configure the NUT's CoAP Ping Server on port 61430.
5. Connect NUT to TN1.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.
8. Repeat steps 2-6 with the following source port configurations. {61440,61445, 61490,61470,61695 }
9. Configure TN1's CoAP Ping Server on port 61430.



10. Configure the NUT's CoAP Ping Server on port 61440.
11. Connect NUT to TN1.
12. TN1 transmits MSDU1 to the NUT.
13. Observe NUT.
14. Repeat steps 8-12 with the following destination port configurations. {61440,61445,61490,61470,61695}

Part C: LOWPAN_GHC UDP Port uncompressed

1. Configure TN1 and the NUT to use EUI-64 addresses.
2. Enable GHC on TN1 and the NUT.
3. Configure TN1's CoAP Ping Server on port 61430.
4. Configure the NUT's CoAP Ping Server on port 61430.
5. Connect NUT to TN1.
6. TN1 transmits MSDU1 to the NUT.
7. Observe NUT.

Observable Results:

Part A

Step 7: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the GHC_UDP header should be set to 01. The source port and the last 8-bits of the destination port should be included inline. Ensure that the UDP payload is compressed using GHC.

Step 13: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the GHC_UDP header should be set to 01. The destination port and the last 8-bits of the source port should be included inline. Ensure that the UDP payload is compressed using GHC.

Part B

Step 7: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the GHC_UDP header should be set to 01. The source port and the last 8-bits of the destination port should be included inline. Ensure that the UDP payload is compressed using GHC.

Step 13: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the GHC_UDP header should be set to 10. The destination port and the last 8-bits of the source port should be included inline. Ensure that the UDP payload is compressed using GHC.

Part C

Step 7: The NUT should respond to MSDU1 with a CoAP Reset message. Bit 5 of the IPHC header should be set to 1. Bits 6 and 7 of the GHC_UDP header should be set to 11. The source port and the destination port should be included inline. Ensure that the UDP payload is compressed using GHC.

Possible Problems:

- None.



Test 6LoWPAN.2.1.3: LOWPAN_GHC Compressed ICMPv6 Encoding

Purpose: To verify that the NUT can correctly encode an ICMPv6 frame using GHC encoding.

References:

- [RFC 7400] – [3.1](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the NUT under both Pure and Hybrid topologies in this test case.

Test Frames	
MSDU1	A valid IPv6 Header containing an ICMPv6 Echo Request. The IPv6 header shall be compressed using IPHC. The ICMP next header will also be compressed using GHC.

Table 65: Test Frames

Procedure:

Part A: LOWPAN_GHC ICMPv6 compressing with EUI-64 Addressing

1. Configure TN1 and NUT to use EUI-64 link-local Address.
2. Enable GHC on TN1 and the NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Part B: LOWPAN_GHC ICMPv6 compressing with 16-bit Short Addressing

1. Configure the PAN coordinator to issue 16-bit Short addresses to TN1 and the NUT.
2. Enable GHC on TN1 and the NUT.
3. Connect NUT to TN1.
4. TN1 transmits MSDU1 to the NUT.
5. Observe NUT.

Observable Results:

Part A

Step 5: The NUT should respond to MSDU1 with ICMPv6 Echo Reply message. Bit 5 of the IPHC header should be set to 1. The ICMPv6 header and its payload should be compressed using GHC.

Part B

Step 5: The NUT should respond to MSDU1 with an ICMPv6 Echo Reply message. Bit 5 of the IPHC header should be set to 1. The ICMPv6 header and its payload should be compressed using GHC.

Possible Problems:

- None.



Group 3: Neighbor Discovery

Scope

The following tests cover the different Neighbor Discovery mechanisms.

Overview

The tests in this group verify that the NUT is capable of exhibiting the correct behavior when using the Neighbor Discovery protocol.



Subsection 1: Host Behavior

Scope

The following tests cover the behavior of the NUT acting as a host when using ND to determine 6LoWPAN Capabilities.

Overview

These tests ensure that the NUT can correctly interpret the 6LoWPAN Capabilities Indication Option.



Test 6LoWPAN.3.1.1: 6LoWPAN Capabilities Indication Option Formatting

Purpose: Verify that the HUT properly includes the 6LoWPAN Capabilities Indication Option (6CIO) when sending a Neighbor Solicitation.

References:

- [RFC 7400] – [3.3](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the HUT using the Hybrid Topology in this test case.

Procedure:

1. Enable GHC on both the HUT and TN1.
2. Connect HUT to TN1.
3. Observe the HUT.

Observable Results:

Step 3: The HUT must send a unicast Neighbor Solicitation to TN1 with the following details:

- An 6LoWPAN Capabilities Indication Option with:
 - Type: 36
 - Length: 1
 - Reserved: 0
 - GHC capable: 1

Possible Problems:

- None.



Subsection 2: Router Behavior

Scope

The following tests cover the behavior of the NUT acting as a router when using ND to determine 6LoWPAN Capabilities.

Overview

These tests ensure that the NUT can correctly interpret the 6LoWPAN Capabilities Indication Option.



Test 6LoWPAN.3.2.1: 6LoWPAN Capabilities Indication Option Formatting

Purpose: Verify that the RUT properly includes the 6LoWPAN Capabilities Indication Option (6CIO) when sending a Router Advertisement.

References:

- [RFC 7400] – [3.3](#)

Test Setup: The network is setup according to [Common Topology](#). [Common Test Setup](#) is performed at the beginning of this test part. The [Common Test Cleanup](#) procedure is performed after each part.

Test the RUT with both Pure and Hybrid Topologies in this test case.

Test Frames	
MSDU1	A valid Router Solicitation Message with a 6LoWPAN Capabilities indication Option indicating support for GHC.

Table 66: Test Frames

Procedure:

1. Enable GHC on both the RUT and TN1.
2. Connect RUT to TN1.
3. TN1 transmits MSDU1.
4. Observe the RUT.

Observable Results:

Step 3: The RUT must send a unicast Router Advertisement to TN1 with the following details:

- An 6LoWPAN Capabilities Indication Option with:
 - Type: 36
 - Length: 1
 - Reserved: 0
 - GHC capable: 1

Possible Problems:

- None.