IPv6 READY

Phase II Test Specification
Core Protocols

Technical Document

Revision 3.6.0

IPv6 Forum **http://www.ipv6forum.org**
Converged Test Specification **http://www.ipv6ready.org**
TAHI Project (Japan)
UNH InterOperability Lab (USA)

MODIFICATION RECORD

Version 0.1	May 1, 2003
Version 0.2	May 9, 2003
Version 0.3	June 3, 2003
Version 1.0	January 28, 2004
Version 2.0	March 3, 2004
Version 2.1	April 9, 2004
Version 2.2	May 25, 2004
Version 2.3	June 15, 2004
Version 2.4	July 30, 2004
Version 2.5	August 31, 2004
Version 2.6	September 8, 2004
Version 2.6.1	September 14, 2004
Version 2.6.2	September 29, 2004 • Added Test v6LC.5.1.4 Part B
Version 2.6.3	October 3, 2004 • v6LC.1.2.14A: changed Address[3]: First 8 octets of TR1's Address
Version 2.6.4	November 10, 2004 • Added Advanced Functionality Test List to the Introduction
Version 3.0	November 19, 2004 • Deleted Test v6LC.4.1.8 Part B • Test v6LC.2.1.6 added Reference ND-Section 6.2.1, seperated Steps 1 and 5 for host and router setup. • Test v6LC.2.2.12, removed (Host Only), added Reference ND-Section 6.2.1, added router configurations in Steps 3 and 8. • Test v6LC.2.2.14, removed (Host Only), added Reference ND-Section 6.2.1, added router configurations in Steps 1 and 6.
Version 3.1	November 22, 2004 • Test v6LC.2.2.14, split in to Part A (Host Only), and Part B (Router Only), to allow for RUT configuration
Version 3.2	December 1, 2004 • Test v6LC.1.1.4, Observable Results Part B: fixed typo to forwarded Echo Request • Test v6LC.1.2.2, Observable Results Part B: fixed typo to Link A
Version 3.3	December 9, 2004 • Test v6LC.5.1.13, Changed to Routers Only. • Test v6LC.2.1.21, Packet A: Source Address= TN1's off-link Global Address. Removed Step 8 in Observable Results. Added Observable Results, Step 8: The HUT MUST not send an Echo Reply to Packet A using TR1 as the first hop. • Test v6LC.2.2.13a,b,c Observable Results: Changed Step 6: …In response to the Echo Request, the HUT MUST not transmit an Echo Reply. Changed Step 17 and 31: The HUT MUST not transmit an Echo Reply. • Test v6LC.2.2.14b Added five seconds to observable results.

Version 3.4	December 15, 2004 • Test v6LC.2.1.21, Observable Results. Added to Step 8: The HUT MUST not transmit multicast NS's with a target set to TR1's link-local address. • Test v6LC.2.2.13a,b,c Observable Results. Added to Step 6, 17, 31: The HUT MUST not transmit multicast NS's with a target set to TR1's link-local address.
Version 3.4.1	January 11, 2005 • Test v6LC.1.1.7b, Observable Results. Changed Pointer field to 0x2e • Removed Test v6LC.1.2.14 Part C • Removed Test v6LC.2.2.8 Part B
Version 3.4.2	March 10, 2005 • Test v6LC.1.3.2, Added Common Test Setup 1.1 • Test v6LC.4.1.4, Changed. Specified size of packets. Second Echo Request is Fragmented.
Version 3.5.0	April 19, 2005 • Test v6LC.4.1.4, Added step for Global address scope. Purpose: changed "link-local" to "on-link" • Test v6LC.2.3.14a,b: Removed Step 7
Version 3.5.1	May 9, 2005 • Test v6LC.2.1.10a, b: Added Steps 4 and 10 • Test v6LC.1.3.1f: Added Steps 11 and 12
Version 3.6.0	June 10, 2005 • Removed Test v6LC2.1.4 Prefix Invalidation (Hosts Only), renumbered section 2, group 1 • Test v6LC2.3.12, changed common test setup to 1.1, added steps 1 through 4.

ACKNOWLEDGEMENTS

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

Principle Authors:

University of New Hampshire- InterOperability Lab
Tahi Project

Commentators:

IRISA-INRIA
TTA/IT Testing Laboratory

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.

The IPv6 Logo Program consists of two phases:

Phase I (Short term period)
In a first stage, the Logo will indicate that the product includes IPv6 mandatory core protocols and can interoperate with other IPv6 equipments. The pragmatic approach of the Logo Committee consists in selecting existing relevant interoperability events and conformance and interoperability test suites.

Phase II (Long term period)
The next stage implies proper care, technical consensus and crystal clear technical references. The IPv6 Ready logo will indicate that a product has successfully satisfied strong requirements stated by the v6LC. To avoid confusion, the IPv6 Ready logo will be generic, and the v6LC will identify different categories of products with associated requirements.

Abbreviations and Acronyms

DAD:	Duplicate Address Detection
HUT:	Host Under Test
MTU:	Maximum Transmission Unit
NCE:	Neighbor Cache Entry
NUT:	Node Under Test
RUT:	Router Under Test
TLLA:	Target Link-layer Address
TN:	Testing Node
TR:	Testing Router

Advanced Functionality Tests

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

Transmitting Echo Requests or configuring packet size:

- v6LC.4.1.10
- v6LC.4.1.11
- v6LC.5.1.1

Multicast Routing:

- v6LC.1.2.6 G, H
- v6LC.1.2.7 G, H
- v6LC.5.1.4 B

MTU Configuration:

- v6LC.5.1.4
- v6LC.5.1.11 B
- v6LC.5.1.12 B
- v6LC.5.1.13 B

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.
Resource Requirements:	The Resource Requirements section specifies the software, hardware, and test equipment that will be needed to perform the test.
Discussion:	The Discussion is a general discussion of the test and relevant section of the specification, including any assumptions made in the design or implementation of the test as well as known limitations.
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 RUT 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 RUT'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.

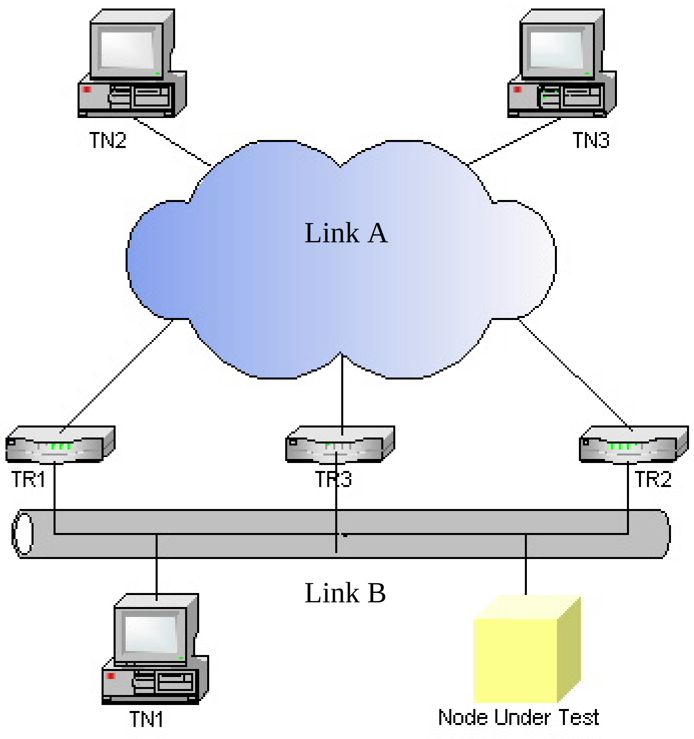
REFERENCES

The following documents are referenced in these texts:

[ADDRCONF]	Thomson, S., T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462, December 1998.
[ICMPv6]	Conta, A., S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 2463, December 1998.
[IPv6-SPEC]	Hinden, R., S. Deering, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.
[ND]	Narten, T., Nordmark, E., and W. Simpson, Neighbor Discovery for IP Version 6 (IPv6), RFC 2461, December 1998.
[PMTU]	McCann, J., S. Deering, and J. Mogul, Path MTU Discovery for IPv6, RFC 1981, August 1996.
[ADDR]	Hinden, R., S. Deering, IP Version 6 Addressing Architecture, RFC 2373, July 1998.
[DS-FIELD]	Nichols, K., S. Blake, F. Baker, and D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, December 1998.
[ECN]	Ramakrishnan, K., S. Floyd, and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168, September 2001.

Common Topology

This topology is used for all tests in this test suite.



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, each TR or TN will respond to Neighbor Solicitations with standard Neighbor Advertisements.

Common Test Setup 1.1

Summary: This minimal setup procedure provides the NUT with a default router TR1, a global prefix, and ensures that the NUT can communicate with TR1.

1. If the NUT is a host, TR1 transmits a Router Advertisement to the all-nodes multicast address. The Router Advertisement includes a Prefix Advertisement with a global prefix and the L and A bits set. This should cause the NUT to add TR1 to its Default Router List, configure a global address, and compute Reachable Time. The Router and Prefix Lifetimes are long enough such that they do not expire during the test.
2. If the NUT is a router, configure a default route with TR1 as the next hop.
3. TR1 transmits an Echo Request to the NUT and responds to Neighbor Solicitations from the NUT. Wait for an Echo Reply from the NUT. This should cause the NUT to resolve the address of TR1 and create a Neighbor Cache entry for TR1 in state REACHABLE.

Common Test Setup 1.2

Summary: This minimal setup procedure provides the NUT with two default routers TR1 and TR2, a global prefix, and ensures that the NUT can communicate with TR1 and TR2.

1. TR1 and TR2 each transmit a Router Advertisement to the all-nodes multicast address. The Router Advertisements include a Prefix Advertisement with a global prefix and the L and A bits set. This should cause the NUT to add TR1 and TR2 to its Default Router List, configure a global address, and compute Reachable Time. The Router and Prefix Lifetimes are long enough such that they do not expire during the test. (If the NUT is a router, configure it to have an address with the advertised prefix.)
2. TR1 and TR2 each transmit an Echo Request to the NUT and respond to Neighbor Solicitations from the NUT. Wait for Echo Replies from the NUT. This should cause the NUT to resolve the addresses of TR1 and TR2 and create a Neighbor Cache entry for each router in state REACHABLE.

Common Test Setup 1.3

Summary: This minimal setup procedure provides the NUT with three default routers TR1, TR2, and TR3, a global prefix, and ensures that the NUT can communicate with TR1, TR2, and TR3.

1. TR1, TR2, and TR3 each transmit a Router Advertisement to the all-nodes multicast address. The Router Advertisements include a Prefix Advertisement with a global prefix and the L and A bits set. This should cause the NUT to add all three routers to its Default Router List, configure a global address, and compute Reachable Time. The Router and Prefix Lifetimes are long enough such that they do not expire during the test.
2. TR1, TR2, and TR3 each transmit an Echo Request to the NUT and respond to Neighbor Solicitations from the NUT. Wait for Echo Replies from the NUT. This should cause the NUT

to resolve the addresses of all three routers and create a Neighbor Cache entry for each router in state REACHABLE.

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.

1. If a TR transmitted a Router Advertisement in the Test Setup or Procedure, that TR transmits a Router Advertisement with the Router Lifetime and each Prefix Lifetime, if applicable, set to zero.
2. Each TR or TN in the test transmits a Neighbor Advertisement for each Neighbor Cache Entry with a Target Link-layer Address Option containing a different cached address. The Override flag should be set.
3. Each TR or TN transmits an Echo Request to the NUT and waits for an Echo Reply.
4. Each TR or TN does not respond to further Neighbor Solicitations.

Common Defaults (for all tests)

Link MTU set to the associated media type default MTU for all nodes on all interfaces.

Section 1: RFC 2460

Scope

The following tests cover the base specification for Internet Protocol version 6, Request For Comments 2460. The base specification specifies the basic IPv6 header and the initially defined IPv6 extension headers and options. It also discusses packet size issues, the semantics of flow labels and traffic classes, and the effects of IPv6 on upper-layer protocols.

Overview

These tests are designed to verify the readiness of an IPv6 implementation vis-à-vis the IPv6 Base specification.

Default Packets

IPv6 Header

Version: 6 Traffic Class: 0 Flow Label: 0 Next Header: 59 (None) Hop Limit: 255 Destination Address: NUT's Link-local Address

Echo Request

IPv6 Header Payload Length: 16 Next Header: 58
ICMPv6 Header Type: 128 Code: 0

Neighbor Advertisement

IPv6 Header Next Header: 58 Destination Address: NUT
Neighbor Advertisement Router flag: 0 for TN1, 1 for TR1 Solicited flag: 1 Override flag: 1 Target Address: TN1/TR1's Link-local Address

Group 1: IPv6 Header

Scope

The following tests cover the fields in the basic IPv6 header.

Overview

Tests in this group verify that a node properly processes and generates the Version, Traffic Class, Flow Label, Payload Length, Next Header, and Hop Limit fields in the IPv6 header. These tests also verify a node transmits the appropriate ICMPv6 Parameter Problem messages in response to invalid or unknown fields.

Test v6LC.1.1.1: Version Field

Purpose: Verify that a node properly processes the Version field of received packets.

References:

- [IPv6-SPEC] – Section 3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Version field of IPv6 packets is a 4-bit field. RFC 2460 does not specify the behavior of an IPv6 node upon reception of a packet with a Version field other than the current version.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

Packet A
IPv6 Header Version: [See below]
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to the NUT, which has an IPv6 header with Version field of 4.
2. TN1 transmits an Echo Request to the NUT.
3. Observe the NUT.
4. Repeat Steps 1 and 2 with a Version field of 0, 5, 7, and 15.

Observable Results:

Step 3: The NUT must not crash or generate invalid packets. In Step 2, the NUT must respond to the second Echo Request from TN1.

Possible Problems:

- None.

Test v6LC.1.1.2: Traffic Class Non-Zero – End Node

Purpose: Verify that a node properly processes the Traffic Class field of received packets and generates a valid value in transmitted packets.

References:

- [IPv6-SPEC] – Section 7
- [DS-FIELD] – Section 3
- [ECN] – Section 5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. The default value must be zero for all 8 bits. Nodes that support a specific (experimental or eventual standard) use of some or all of the Traffic Class bits are permitted to change the value of those bits in packets that they originate, forward, or receive, as required for that specific use. Nodes should ignore and leave unchanged any bits of the Traffic Class field for which they do not support a specific use.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Traffic Class: 32 Next Header: 58
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to the NUT, an Echo Request with a Traffic Class field of 32, which is non-zero.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must generate an Echo Reply. If the NUT supports a specific use of the Traffic Class field, the Traffic Class in the Echo Reply may be non-zero. Otherwise, the Traffic Class field should be zero.

Possible Problems:

- None.

Test v6LC.1.1.3: Traffic Class Non-Zero – Intermediate Node (Routers Only)

Purpose: Verify that a router properly processes the Traffic Class field of received packets and generates a valid value in transmitted packets.

References:

- [IPv6-SPEC] – Section 7
- [DS-FIELD] – Section 3
- [ECN] – Section 5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. The default value must be zero for all 8 bits. Nodes that support a specific (experimental or eventual standard) use of some or all of the Traffic Class bits are permitted to change the value of those bits in packets that they originate, forward, or receive, as required for that specific use. Nodes should ignore and leave unchanged any bits of the Traffic Class field for which they do not support a specific use.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

1. Enable the RUT's interface on Link A.

Packet A

IPv6 Header Traffic Class: 32 Next Header: 58
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to TN2's Global Address with a first hop through the RUT, an Echo Request with a Traffic Class field of 32, which is non-zero.
2. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT must forward the Echo Request. If the RUT supports a specific use of the Traffic Class field, the Traffic Class in the Echo Request may be non-zero. Otherwise, the Traffic Class field should be passed on to TN2 unchanged.

Possible Problems:

- None.

Test v6LC.1.1.4: Flow Label Non-Zero

Purpose: Verify that a node properly processes the Flow Label field of received packets and generates a valid value in transmitted packets.

References:

- [IPv6-SPEC] – Section 6, Appendix A

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet.

A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Flow Label: 214375 Next Header: 58
ICMPv6 Echo Request

Procedure:

Part A: NUT receives Non-Zero Flow Label

1. TN1 transmits Packet A, an Echo Request with a Flow Label of 0x34567 to the NUT.
2. Observe the packets transmitted by the NUT.

Part B: RUT forwards Non-Zero Flow Label (Routers Only)

3. On the RUT, enable its interface to Link A.
4. Configure the RUT to advertise a different prefix on Link B and Link A.
5. TN1 transmits Packet A, an Echo Request with a Flow Label 0x34567 to TN2’s Global address with a first hop through the RUT.
6. Observe the packets transmitted by the RUT on Link A.

Observable Results:

- *Part A*

  Step 2: The NUT must generate an Echo Reply. If the NUT supports use of the Flow Label field, the Flow Label in the Echo Reply may be non-zero. Otherwise, the Flow Label field must be zero.
- *Part B*

  Step 6: The RUT must forward the Echo Request from TN1 to TN2. If the RUT does not support the use of the Flow Label field, it must be unchanged in the forwarded packet.

Possible Problems:

- None.

Test v6LC.1.1.5: Payload Length

Purpose: Verify that a node properly processes the Payload Length field of received packets.

References:

- [IPv6-SPEC] – Section 3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: An IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header. The Payload Length indicates the length of the IPv6 payload, i.e., the rest of the packet following the IPv6 header, in octets. RFC 2460 does not specify the behavior of an IPv6 node upon reception of a packet with an invalid Payload Length.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Payload Length: [See below] Next Header: 58
ICMPv6 Echo Request

Procedure:

Part A: Payload Length Odd

1. TN1 transmits Packet A to the NUT, an Echo Request that has an IPv6 header with a Payload Length of 0x33 (51).
2. Observe the packets transmitted by the NUT.

Part B: Payload Length Even

3. TN1 transmits Packet A to the NUT, an Echo Request that has an IPv6 header with a Payload Length of 0x32 (50).
4. Observe the packets transmitted by the NUT.

Observable Results:

- *Parts A and B*
  Steps 2 and 4: The NUT must generate an Echo Reply, indicating successful processing of the packet.

Possible Problems:

- None.

Test v6LC.1.1.6: No Next Header after IPv6 Header

Purpose: Verify proper behavior of a node when it encounters a Next Header value of 59 (no next header).

References:

- [IPv6-SPEC] – Section 4.7

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The value 59 in the Next Header field of an IPv6 Header or any extension header indicates that there is nothing following that header. If the Payload Length field of the IPv6 Header indicates the presence of octets past the end of a header whose Next Header field contains 59, those octets must be ignored, and passed on unchanged if the packet is forwarded.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 59
ICMPv6 Echo Request

Procedure:

Part A: NUT Receives No Next Header

1. TN1 transmits Packet A to the NUT, which contains an IPv6 header with a Next Header of 59. Following the IPv6 header is an ICMPv6 Echo Request Header.
2. Observe the NUT.

Part B: RUT Forwards No Next Header – (Routers Only)

3. On the RUT, enable its interface to Link A.
4. Configure the RUT to advertise a different prefix on Link B and Link A.
5. TN1 transmits Packet A, an Echo Request containing an IPv6 header with a Next Header of 59 to TN2's Global address with a first hop through the RUT.
6. Observe the packets transmitted by the RUT on Link A.

Observable Results:

- *Part A*

  Step 2: The NUT must not send any packets in response to Packet A.
- *Part B*

Step 6: The RUT must forward Packet A to TN2. The octets after the IPv6 header with a Next Header field of 59 (the ICMPv6 Request octets) must be unchanged.

Possible Problems:

- None.

Test v6LC.1.1.7: Unrecognized Next Header

Purpose: Verify that a node generates the appropriate response to an unrecognized or unexpected Next Header field.

References:

- [IPv6-SPEC] – Section 4
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If, as a result of processing a header, a node is required to proceed to the next header, but the Next Header value in the current header is unrecognized by the node, it should discard the packet and transmit an ICMPv6 Parameter Problem message to the source of the packet. The ICMPv6 Code value should be 1 ("unrecognized Next Header type encountered.") The ICMPv6 Pointer field should contain the offset of the unrecognized value within the original packet.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A
IPv6 Header Next Header: [See below]

Packet B
IPv6 Header Next Header: 0
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 0 ID: 135
ICMPv6 Echo Request

Procedure:

Part A: Unrecognized Next Header in IPv6 Header (Multiple Values)

1. TN1 transmits Packet A to the NUT, which has an IPv6 header with a Next Header field of 136.

2. TN1 transmits a valid Echo Request to the NUT.
3. Observe the packets transmitted by the NUT.
4. Repeat Steps 1 and 2 with all unrecognized Next Header values between 137 and 254 in Step 1.

Part B: Unexpected Next Header in IPv6 Header

5. TN1 transmits Packet B to the NUT, which has an IPv6 header with a Next Header field of 0. The actual extension header that follows is a Fragment header. The Fragment ID is 135.
6. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 3: The NUT should send an ICMPv6 Parameter Problem message to TN1. The ICMPv6 Code field should be 1 (unrecognized Next Header type encountered). The ICMPv6 Pointer field should be 0x06 (offset of the Next Header field). The NUT must respond to the Echo Request from TN1 in step 2.
- *Part B*

  Step 6: The NUT would interpret the Fragment header as a Hop-by-Hop Options header. Thus, the Fragment ID would be interpreted as if it were an Option Type The NUT should send an ICMPv6 Parameter Problem message to TN1. The Code field should be 2 (unrecognized IPv6 Option encountered). The Pointer field should be 0x2e (offset of the Fragment ID in the Fragment header). The NUT should discard Packet B and should not send an Echo Reply to TN1.

Possible Problems:

- None.

Test v6LC.1.1.8: Hop Limit Zero – End Node

Purpose: Verify that a node correctly processes the Hop Limit field of received packets and generates a valid value in transmitted packets.

References:

- [IPv6-SPEC] – Sections 3 and 8.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Hop Limit field is decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. Unlike IPv4, IPv6 nodes are not required to enforce maximum packet lifetime. That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6. In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a change in practice.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A
IPv6 Header Hop Limit: 0 Next Header: 58
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to the NUT, an Echo Request with a Hop Limit field of zero.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must generate an Echo Reply with a Hop Limit field value of greater than zero.

Possible Problems:

- None.

Test v6LC.1.1.9: Hop Limit Decrement – Intermediate Node (Routers Only)

Purpose: Verify that a router correctly processes the Hop Limit field of received packets and generates a valid value in transmitted packets.

References:

- [IPv6-SPEC] – Sections 3 and 8.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Hop Limit field is decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. Unlike IPv4, IPv6 nodes are not required to enforce maximum packet lifetime. That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6. In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a change in practice.

The reception of Hop Limit field equal to zero and Hop Limit field decremented to zero is tested in the ICMPv6 section of this test suite.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A
IPv6 Header Hop Limit: 15 Next Header: 58
ICMPv6 Echo Request

Procedure:

1. On the RUT, enable its interface to Link A.
2. Configure the RUT to advertise different prefixes on Link A and Link B.
3. TN1 transmits Packet A to TN2’s Global Address with a first hop through the RUT. The Hop Limit field is set to 15.
4. Observe the packets transmitted by the RUT on Link A.

Observable Results:

Step 12: The RUT should forward Packet A to TN2. The Hop Limit field should be decremented to 14.

Possible Problems:

- None.

Group 2: Extension Headers and Options

Scope

The following tests cover the processing of options and extension headers, particularly the Hop-by-Hop Options, Destination Options, and Routing headers.

Overview

Tests in this group verify that a node properly processes and generates the Header Extension Length field in extension headers, and the Option Type and Option Data Length fields in IPv6 options. These tests also verify that a node correctly processes header options in order, packets with a routing header destined for the node, and many extension headers or options in a single packet. In addition, these tests ensure a node generates the proper ICMPv6 message in response to invalid or unknown fields.

Test v6LC.1.2.1: Next Header Zero

Purpose: Verify that a node discards a packet that has a Next Header field of zero in a header other than an IPv6 header and generates an ICMPv6 Parameter Problem message to the source of the packet.

References:

- [IPv6-SPEC] – Section 4
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The value of zero in the Next Header field of the IPv6 header indicates the presence of the Hop-by-Hop Options header. When present, the Hop-by-Hop Options header must immediately follow the IPv6 header. The Hop-by-Hop Options header contains information that must be examined by every node along a packet's delivery path, including the source and destination nodes. As a result of processing a header, if a node encounters a Next Header value of zero in any header other than an IPv6 header, it should discard the packet and transmit an ICMPv6 Parameter Problem message to the source of the packet. The ICMPv6 Code field should be equal to 1 ("unrecognized Next Header type encountered") and the ICMPv6 Pointer field contains the offset of the unrecognized value within the original packet.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 0
Hop-by-Hop Options Header Next Header: 0 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
Hop-by-Hop Options Header Next Header: 58 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to the NUT, which has a Hop-by-Hop Options header with a Next Header field of zero.

2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT should send an ICMPv6 Parameter Problem message to TN1. The ICMPv6 Code field should be 1 (unrecognized Next Header type encountered). The ICMPv6 Pointer field should be 0x28 (offset of the Next Header field of the Hop-by-Hop Options header). The NUT should discard the Echo Request and not send an Echo Reply to TN1.

Possible Problems:

- None.

Test v6LC.1.2.2: No Next Header after Extension Header

Purpose: Verify proper behavior of a node when it encounters a Next Header value of 59 (no next header).

References:

- [IPv6-SPEC] – Section 4.7

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The value 59 in the Next Header field of an IPv6 Header or any extension header indicates that there is nothing following that header. If the Payload Length field of the IPv6 Header indicates the presence of octets past the end of a header whose Next Header field contains 59, those octets must be ignored, and passed on unchanged if the packet is forwarded.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 60
Destination Options Header Next Header: 59 (None) Header Ext. Length: 0 Option: PadN Opt Data Len: 4
ICMPv6 Echo Request

Procedure:

Part A: End Node

1. TN1 transmits Packet A to the NUT, which contains a Destination Options header with a Next Header of 59. Following the Destination Options header is an ICMPv6 Echo Request header.
2. Observe the packets transmitted by the NUT.

Part B: Intermediate Node (Routers Only)

3. On the RUT, enable its interface on Link A.
4. Configure the RUT to advertise different prefixes on Link A and Link B.
5. TN1 transmits Packet A to TN2 with a first hop through the RUT. Packet A contains a Destination Options header with a Next Header of 59. Following the Destination Options header is an ICMPv6 Echo Request header.
6. Observe the packets transmitted by the RUT on Link A.

Observable Results:

- *Part A*
  Step 2: The NUT must not send any packets in response to Packet A.
- *Part B*
  Step 6: The RUT should forward Packet A to TN2 on Link A. The octets past the end of the header whose Next Header field contains 59 must be unchanged.

Possible Problems:

- None.

Test v6LC.1.2.3: Unrecognized Next Header in Extension Header – End Node

Purpose: Verify that a node discards a packet with an unrecognized or unexpected next header in an extension header and transmits an ICMPv6 Parameter Problem message to the source of the packet.

References:

- [IPv6-SPEC] – Section 4
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If, as a result of processing a header, a node is required to proceed to the next header, but the Next Header value in the current header is unrecognized by the node, it should discard the packet and transmit an ICMPv6 Parameter Problem message to the source of the packet. The ICMPv6 Code value should be 1 ("unrecognized Next Header type encountered.") The ICMPv6 Pointer field should contain the offset of the unrecognized value within the original packet. The same action should be taken if a node encounters a Next Header value of zero in any header other than the IPv6 header.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 60
Destination Options Header Next Header: [See below] Header Ext. Length: 0 Option: PadN Opt Data Len: 4

Packet B

IPv6 Header Next Header: 60
Destination Options Header Next Header: 60 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
Fragment Header

Next Header: 58 Reserved: 0 Fragment Offset: 0x10E0 (First 8 bits = 135) Res: 0x2 More Fragments flag: 0
ICMPv6 Echo Request

Procedure:

Part A: Unrecognized Next Header in Extension Header (Multiple Values)

1. TN1 transmits Packet A, which has a Destination Options header with a Next Header field of 136.
2. TN1 transmits a valid Echo Request to the NUT.
3. Repeat Steps 1 and 2 with all unrecognized Next Header values between 137 and 254 in Step 1.
4. Observe the Packets transmitted by the NUT.

Part B: Unexpected Next Header in Extension Header

5. TN1 transmits Packet B, which has a Destination Options header with a Next Header field of 60. The actual extension header that follows is a Fragment header. The Fragment Offset is 0x10E0 (so that the first 8 bits of this 13 bit field would be 135). The second reserved field is 0x2 and the more bit is clear. (If processed as a Destination Options header, this would be processed as Option Data Length equals 4.)
6. Observe the Packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 4: The NUT should send an ICMPv6 Parameter Problem message to TN1. The ICMPv6 Code field should be 1 (unrecognized Next Header type encountered). The ICMPv6 Pointer field should be 0x28 (offset of the Next Header field). The NUT should send an Echo Reply in response to the Echo Request sent by TN1 in Step 2.
- *Part B*

  Step 6: From the Next Header field in the Destination Options header, the NUT expects the Fragment header to be a Destination Options header. Thus, the Fragment Offset would be interpreted as if it were an Option Type. The NUT should send an ICMPv6 Parameter Problem message to TN1. The Code field should be 2 (unrecognized IPv6 Option encountered). The Pointer field should be 0x32 (offset of the Fragment Offset in the Fragment header). The NUT should discard Packet B and should not send an Echo Reply to TN1.

Possible Problems:

- None.

Test v6LC.1.2.4: Extension Header Processing Order

Purpose: Verify that a node properly processes the headers of an IPv6 packet in the correct order.

References:

- [IPv6-SPEC] – Sections 4, 4.1, 4.2, and 4.5
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet. The exception is the Hop-By-Hop Options header, which is restricted to appear immediately after an IPv6 header only. The sequence of options within a header must be processed strictly in the order they appear in the header. The Option Type identifiers are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type. Each extension header is an integer multiple of 8 octets long, in order to retain 8-octet alignment for subsequent headers.

Option Type Identifiers (highest order two bits):

- 00_b: Skip over this option and continue processing the header
- 01_b: Discard the packet
- 10_b: Discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and an ICMPv6 Pointer field pointing to the unrecognized Option Type.
- 11_b: Discard the packet and, only if the packet's Destination Address was not a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and an ICMPv6 Pointer field pointing to the unrecognized Option Type.

If the length of a fragment, as derived from the fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is 1, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Payload Length field of the fragment packet.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header Next Header: 0 Payload Length: 37	IPv6 Header Next Header: 0 Payload Length: 37
Hop-by-Hop Options Header Next Header: 60 Header Ext. Length: 0 Option: PadN Opt Data Len: 4	Hop-by-Hop Options Header Next Header: 60 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
Destination Options Header Next Header: 44 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4	Destination Options Header Next Header: 44 Header Ext. Length: 0 Option: 7 (unknown, msb: 00_b) Opt Data Len: 4
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1	Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1
ICMPv6 Echo Request Data Length: 5	ICMPv6 Echo Request Data Length: 5

Packet C	Packet D
IPv6 Header Next Header: 0 Payload Length: 37	IPv6 Header Next Header: 0 Payload Length: 37
Hop-by-Hop Options Header Next Header: 44 Header Ext. Length: 0 Option: PadN Opt Data Len: 4	Hop-by-Hop Options Header Next Header: 44 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
Fragment Header Next Header: 60 Fragment Offset: 0 More Fragments flag: 1	Fragment Header Next Header: 60 Fragment Offset: 0 More Fragments flag: 0
Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4	Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4
ICMPv6 Echo Request Data Length: 5	ICMPv6 Echo Request Data Length: 5

Procedure:

Part A: Destination Options Header precedes Fragment Header, Error from Destination Options Header

1. TN1 transmits Packet A, an Echo Request that has a Hop-by-Hop Options header, Destination Options header, and Fragment header, in that order. The Destination Options header has an

unknown Option Type of 135. The IPv6 header has a Payload Length that is not a multiple of 8 octets, and the Fragment header has the M-bit set.

2. Observe the packets transmitted by the NUT.

Part B: Destination Options Header precedes Fragment Header, Error from Fragment Header

3. TN1 transmits Packet B, an Echo Request that has a Hop-by-Hop Options header, Destination Options header, and Fragment header, in that order. The Destination Options header has an unknown Option Type of 7. The IPv6 header has a Payload Length that is not a multiple of 8 octets, and the Fragment header has the M-bit set.
4. Observe the packets transmitted by the NUT.

Part C: Fragment Header precedes Destination Options Header, Error from Fragment Header

5. TN1 transmits Packet C, an Echo Request that has a Hop-by-Hop Options header, Fragment header, and Destination Options header, in that order. The IPv6 header has a Payload Length that is not a multiple of 8 octets, and the Fragment header has the M-bit set. The Destination Options header has an unknown Option Type of 135.
6. Observe the packets transmitted by the NUT.

Part D: Fragment Header precedes Destination Options Header, Error from Destination Options Header

7. TN1 transmits Packet D, an Echo Request that has a Hop-by-Hop Options header, Fragment header, and Destination Options header, in that order. The IPv6 header has a Payload Length that is not a multiple of 8 octets, and the Fragment header does not have the M-bit set. The Destination Options header has an unknown Option Type of 135.
8. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x32 (offset of the Option type field in the Destination Options header). The NUT must discard the Echo Request from TN1.
- *Part B*

  Step 4: The NUT should send an ICMPv6 Parameter Problem message to TN1. The Code field should be 0 (erroneous header field encountered). The Pointer field should be 0x04 (offset of the Payload Length field in the IPv6 header). The NUT must discard the Echo Request from TN1.
- *Part C*

  Step 6: NUT should send an ICMPv6 Parameter Problem message to TN1. The Code field should be 0 (erroneous header field encountered). The Pointer field should be 0x04 (offset of the Payload Length field in the IPv6 header). The NUT must discard the Echo Request from TN1.
- *Part D*

  Step 8: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). If the IPv6 Parameter Problem message includes a Fragment Header, the Pointer field must be 0x3A (offset of the Option type field in the Destination Options header). If the IPv6 Parameter Problem message does not include a Fragment Header, the Pointer field must be 0x32 (offset of the Option type field in the Destination Options header). The NUT must discard the Echo Request from TN1.

Possible Problems:

- None.

Test v6LC.1.2.5: Option Processing Order

Purpose: Verify that a node properly processes the options in a single header in the order of occurrence.

References:

- [IPv6-SPEC] – Section 4.2
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The sequence of options within a header must be processed strictly in the order they appear in the header; a receiver must not, for example, scan through the header looking for a particular kind of option and process that option prior to processing all preceding ones. The Option Type identifiers are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

- 00_b: Skip over this option and continue processing the header.
- 01_b: Discard the packet.
- 10_b: Discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.
- 11_b: Discard the packet and, only if the packet's Destination Address was not a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header Next Header: 60	IPv6 Header Next Header: 60

Destination Options Header Next Header: 58 Header Ext. Length: 3 Option: 7 (unknown, msb: 00_b) Opt Data Len: 4 Option: 71 (unknown, msb: 01_b) Opt Data Len: 6 Option: 135 (unknown, msb: 10_b) Opt Data Len: 6 Option: 199 (unknown, msb: 11_b) Opt Data Len: 6	Destination Options Header Next Header: 58 Header Ext. Length: 3 Option: 7 (unknown, msb: 00_b) Opt Data Len:: 4 Option: 135 (unknown, msb: 10_b) Opt Data Len: 6 Option: 199 (unknown, msb: 11_b) Opt Data Len: 6 Option: 71 (unknown, msb: 01_b) Opt Data Len: 6
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet C

IPv6 Header Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 3 Option: 7 (unknown, msb: 00_b) Opt Data Len: 4 Option: 199 (unknown, msb: 11b) Opt Data Len: 6 Option: 71 (unknown, msb: 01_b) Opt Data Len: 6 Option: 135 (unknown, msb: 10b) Opt Data Len: 6
ICMPv6 Echo Request

Procedure:

Part A: First Option has Most Significant Bits 00_b, Next has Most Significant Bits 01_b

1. TN1 transmits Packet A to the NUT, an Echo Request that has a Destination Options header with four unknown Options. The Option Types are 7, 71, 135, and 199.
2. Observe the NUT.

Part B: First Option has Most Significant Bits 00_b, Next has Most Significant Bits 10_b

3. TN1 transmits Packet B to the NUT, an Echo Request that has a Destination Options header with four unknown Options. The Option Types are 7, 135, 199, and 71.
4. Observe the packets transmitted by the NUT.

Part C: First Option has Most Significant Bits 00_b, Next has Most Significant Bits 11_b

5. TN1 transmits Packet C to the NUT's link-local address, an Echo Request that has a Destination Options header with four unknown Options. The Option Types are 7, 199, 71, and 135.
6. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

Step 2: The NUT must silently discard the ICMPv6 Echo Request and not send any packets to TN1.

- *Part B*

  Step 4: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x30 (offset of the Option Type field of the second option). The NUT must discard the Echo Request sent by TN1 and must not send a Reply.
- *Part C*

  Step 6: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x30 (offset of the Option Type field of the second option). The NUT must discard the Echo Request sent by TN1 and must not send a Reply.

Possible Problems:

- None.

Test v6LC.1.2.6: Options Processing, Hop-by-Hop Options Header - End Node

Purpose: Verify that a node properly processes both known and unknown options, and acts in accordance with the highest order two bits of the option.

References:

- [IPv6-SPEC] – Sections 4.2 and 4.3
- [ICMPv6] – Sections 2.2, 2.4 and 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Option Type identifiers are internally encoded such that their highest order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

- 00_b: Skip over this option and continue processing the header.
- 01_b: Discard the packet.
- 10_b: Discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.
- 11_b: Discard the packet and, only if the packet's Destination Address was not a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.

The pointer field identifies the octet of the original packet's header where the error was detected.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: Pad1	Option: PadN
Option: Pad1	Opt Data Len: 4
Option: Pad1	
Option: Pad1	
Option: Pad1	
Option: Pad1	
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet C	Packet D
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: 7 (unknown, msb: 00_b)	Option: 71 (unknown, msb: 01_b)
Opt Data Len: 4 bytes	Opt Data Len: 4 bytes
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet E	Packet F
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: 135 (unknown, msb: 10_b)	Option: 199 (unknown, msb: 11_b)
Opt Data Len: 4 bytes	Opt Data Len: 4 bytes
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet G	Packet H
IPv6 Header	IPv6 Header
Destination Address: All Nodes	Destination Address: All Nodes
Link-local Multicast	Link-local Multicast
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: 135 (unknown, msb: 10_b)	Option: 199 (unknown, msb: 11_b)
Opt Data Len: 4 bytes	Opt Data Len: 4 bytes
ICMPv6 Echo Request	ICMPv6 Echo Request

Procedure:

Part A: Pad1 Option

1. TN1 transmits Packet A to the NUT, an Echo Request that has a Hop-by-Hop Options header with six Pad1 Options.
2. Observe the packets transmitted by the NUT.

Part B: PadN Option

3. TN1 transmits Packet B to the NUT, an Echo Request that has a Hop-by-Hop Options header with a 6 byte PadN Option.
4. Observe the packets transmitted by the NUT.

Part C: Most Significant Bits 00_b

5. TN1 transmits Packet C to the NUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 7.
6. Observe the packets transmitted by the NUT.

Part D: Most Significant Bits 01_b

7. TN1 transmits Packet D to the NUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 71.
8. Observe the NUT.

Part E: Most Significant Bits 10_b, unicast destination

9. TN1 transmits Packet E to the NUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 135.
10. Observe the packets transmitted by the NUT.

Part F: Most Significant Bits 11_b, unicast destination

11. TN1 transmits Packet F to the NUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 199.
12. Observe the packets transmitted by the NUT.

Part G: Most Significant Bits 10_b, multicast destination

13. TN1 transmits Packet G, an Echo Request sent to a local multicast address that has a Hop-by-Hop Options header with an unknown Option Type of 135.
14. Observe the packets transmitted by the NUT.

Part H: Most Significant Bits 11_b, multicast destination

15. TN1 transmits Packet H, an Echo Request sent to a local multicast address that has a Hop-by-Hop Options header with an unknown Option Type of 199.
16. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must send an Echo Reply to TN1.
- *Part B*

  Step 4: The NUT must send an Echo Reply to TN1.
- *Part C*

  Step 6: The unknown option is skipped and the header is processed. The NUT must send an Echo Reply to TN1.
- *Part D*

  Step 8: The NUT must not generate any packets sent to TN1. The Echo Request is discarded.

- *Part E*

  Step 10: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Source Address of the Parameter Problem Message must be the same as the Destination Address in TN1's Echo Request Packet.
    - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part F*

  Step 12: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Source Address of the Parameter Problem Message must be the same as the Destination Address in TN1's Echo Request Packet.
    - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part G*

  Step 14: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Destination Address of the Parameter Problem Message should be the same as the Source Address in TN1's Echo Request Packet.
- *Part H*

  Step 16: The NUT must not generate any packets sent to TN1. The Echo Request is discarded, as the destination address is multicast. The NUT must not send an ICMPv6 Parameter Problem message.

Possible Problems:

- None.

Test v6LC.1.2.7: Options Processing, Hop-by-Hop Options Header - Intermediate Node (Routers Only)

Purpose: Verify that a router properly processes both known and unknown options, and acts in accordance with the highest order two bits of the option.

References:

- [IPv6-SPEC] – Sections 4.2 and 4.3
- [ICMPv6] – Sections 2.2, 2.4 and 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: With one exception, extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node identified in the Destination Address field of the IPv6 header.

The exception referred to above is the Hop-by-Hop Options header, which carries information that must be examined and processed by every node along a packet's delivery path, including the source and destination nodes. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

Option Type identifiers are internally encoded such that their highest order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

- 00_b: Skip over this option and continue processing the header.
- 01_b: Discard the packet.
- 10_b: Discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.
- 11_b: Discard the packet and, only if the packet's Destination Address was not a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.

The pointer field identifies the octet of the original packet's header where the error was detected.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part. Enable the RUT's interface to Link A.

Packet A	Packet B
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: Pad1	Option: PadN
Option: Pad1	Opt Data Len: 4
Option: Pad1	
Option: Pad1	
Option: Pad1	
Option: Pad1	
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet C	Packet D
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: 7 (unknown, msb: 00_b)	Option: 71 (unknown, msb: 01_b)
Opt Data Len: 4	Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet E	Packet F
IPv6 Header	IPv6 Header
Next Header: 0	Next Header: 0
Hop-by-Hop Options Header	Hop-by-Hop Options Header
Next Header: 58	Next Header: 58
Header Ext. Length: 0	Header Ext. Length: 0
Option: 135 (unknown, msb: 10_b)	Option: 199 (unknown, msb: 11_b)
Opt Data Len: 4	Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet G	Packet H

IPv6 Header Destination Address: Global Scope Multicast TR1 Next Header: 0	IPv6 Header Destination Address: Global Scope Multicast TR1 Next Header: 0
Hop-by-Hop Options Header Next Header: 58 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4	Hop-by-Hop Options Header Next Header: 58 Header Ext. Length: 0 Option: 199 (unknown, msb: 11_b) Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Procedure:

Part A: Pad1 Option

1. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with six Pad1 Options.
2. Observe the packets transmitted by the RUT.

Part B: PadN Option

3. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with a 4 byte PadN Option.
4. Observe the packets transmitted by the RUT.

Part C: Most Significant Bits 00_b

5. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 7.
6. Observe the packets transmitted by the RUT.

Part D: Most Significant Bits 01_b

7. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 71.
8. Observe the RUT.

Part E: Most Significant Bits 10_b, unicast destination

9. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 135.
10. Observe the packets transmitted by the RUT.

Part F: Most Significant Bits 11_b, unicast destination

11. TN1 transmits Packet A to TN2 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 199.
12. Observe the packets transmitted by the RUT.

Part G: Most Significant Bits 10_b, on-link multicast destination

13. TN1 transmits Packet A to the global scope multicast destination of TR1 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 135.
14. Observe the packets transmitted by the RUT.

Part H: Most Significant Bits 11_b, on-link multicast destination

15. TN1 transmits Packet A to the global scope multicast destination of TR1 with a first hop through the RUT, an Echo Request that has a Hop-by-Hop Options header with an unknown Option Type of 199.

16. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*
  Step 2: The RUT must forward the Echo Request to TN2.
- *Part B*
  Step 4: The RUT must forward the Echo Request to TN2.
  Part C
  Step 6: The unknown option is skipped and the header is processed. The RUT must forward the Echo Request to TN2.
- *Part D*
  Step 8: The RUT must not forward the Echo Request to TN2. The Echo Request is discarded.
- *Part E*
  Step 10: The RUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The RUT must discard the Echo Request and not forward it to TN2. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
  - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part F*
  Step 12: The RUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The RUT must discard the Echo Request and not forward it to TN2. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
  - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part G*
  Step 14: The RUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Hop-by-Hop Options header). The RUT must discard the Echo Request and not forward it to TR1. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
  - The Destination Address of the Parameter Problem Message should be the same as the Source Address in TN1's Echo Request Packet.
- *Part H*
  Step 16: The RUT must not forward the Echo Request to TR1. The Echo Request is discarded, as the destination address is multicast. The RUT must not send an ICMPv6 Parameter Problem message.

Possible Problems:

- Part G and H: These tests may be omitted if the RUT does not support Multicast Routing.

Test v6LC.1.2.8: Option Processing, Destination Options Header

Purpose: Verify that a node properly processes both known and unknown options, and acts in accordance with the highest order two bits of the option.

References:

- [IPv6-SPEC] – Sections 4.2 and 4.6
- [ICMPv6] – Sections 2.2, 2.4 and 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Option Type identifiers are internally encoded such that their highest order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

- 00_b: Skip over this option and continue processing the header.
- 01_b: Discard the packet.
- 10_b: Discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.
- 11_b: Discard the packet and, only if the packet's Destination Address was not a multicast address, transmit an ICMPv6 Parameter Problem message to the packet's Source Address with an ICMPv6 Code field of 2 and the ICMPv6 Pointer field containing the offset of the unrecognized Option Type.

The pointer field identifies the octet of the original packet's header where the error was detected.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header Next Header: 60	IPv6 Header Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: Pad1 Option: Pad1 Option: Pad1 Option: Pad1 Option: Pad1 Option: Pad1	Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet C	Packet D
IPv6 Header Next Header: 60	IPv6 Header Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 7 (unknown, msb: 00_b) Opt Data Len: 4	Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 71 (unknown, msb: 01_b) Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet E	Packet F
IPv6 Header Next Header: 60	IPv6 Header Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4	Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 199 (unknown, msb: 11_b) Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Packet G	Packet H
IPv6 Header Destination Address: All Nodes Link-local Multicast Next Header: 60	IPv6 Header Destination Address: All Nodes Link-local Multicast Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 135 (unknown, msb: 10_b) Opt Data Len: 4	Destination Options Header Next Header: 58 Header Ext. Length: 0 Option: 199 (unknown, msb: 11_b) Opt Data Len: 4
ICMPv6 Echo Request	ICMPv6 Echo Request

Procedure:

Part A: Pad1 Option

1. TN1 transmits Packet A to the NUT an Echo Request that has a Destination Options header with six Pad1 Options.
2. Observe the packets transmitted by the NUT.

Part B: PadN Option

3. TN1 transmits Packet B to the NUT, an Echo Request that has a Destination Options header with a 4 byte PadN Option.
4. Observe the packets transmitted by the NUT.

Part C: Most Significant Bits 00_b

5. TN1 transmits Packet C to the NUT, an Echo Request that has a Destination Options header with an unknown Option Type of 7.
6. Observe the packets transmitted by the NUT.

Part D: Most Significant Bits 01_b

7. TN1 transmits Packet D to the NUT, an Echo Request that has a Destination Options header with an unknown Option Type of 71.
8. Observe the NUT.

Part E: Most Significant Bits 10_b, unicast destination

9. TN1 transmits Packet E to the NUT, an Echo Request that has a Destination Options header with an unknown Option Type of 135.
10. Observe the packets transmitted by the NUT.

Part F: Most Significant Bits 11_b, unicast destination

11. TN1 transmits Packet F to the NUT, an Echo Request that has a Destination Options header with an unknown Option Type of 199.
12. Observe the packets transmitted by the NUT.

Part G: Most Significant Bits 10_b, multicast destination

13. TN1 transmits Packet G, an Echo Request sent to a local multicast address that has a Destination Options header with an unknown Option Type of 135.
14. Observe the packets transmitted by the NUT.

Part H: Most Significant Bits 11_b, multicast destination

15. TN1 transmits Packet H, an Echo Request sent to a local multicast address that has a Destination Options header with an unknown Option Type of 199.
16. Observe the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must send an Echo Reply to TN1.
- *Part B*

  Step 4: The NUT must send an Echo Reply to TN1.
- *Part C*

  Step 6: The unknown option is skipped and the header is processed. The NUT must send an Echo Reply to TN1.
- *Part D*

  Step 8: The NUT must not generate any packets sent to TN1. The Echo Request is discarded.

- *Part E*

  Step 10: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Destination Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Source Address of the Parameter Problem Message must be the same as the Destination Address in TN1's Echo Request Packet.
    - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part F*

  Step 12: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Destination Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Source Address of the Parameter Problem Message must be the same as the Destination Address in TN1's Echo Request Packet.
    - The Destination Address should be the same as the Source Address in TN1's Echo Request Packet.
- *Part G*

  Step 14: The NUT must send an ICMPv6 Parameter Problem message to TN1. The Code field must be 2 (unrecognized IPv6 Option encountered). The Pointer field must be 0x2A (offset of the option field of Destination Options header). The NUT must discard the Echo Request and not send a Reply. The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
    - The Destination Address of the Parameter Problem Message should be the same as the Source Address in TN1's Echo Request Packet.
- *Part H*

  Step 16: The NUT must not generate any packets sent to TN1. The Echo Request is discarded, as the destination address is multicast. The NUT must send an ICMPv6 Parameter Problem message. The NUT must discard the Echo Request and not send a Reply.

Possible Problems:

- None.

Test v6LC.1.2.9: Responding to Routing Header - End Node

Purpose: Verify that a node properly responds to an IPv6 packet destined for it that contains a Routing header.

References:

- [IPv6-SPEC] – Sections 4.4 and 8.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination. When an upper-layer protocol sends one or more packets in response to a received packet that included a Routing header, the response packet(s) must not include a Routing header that was automatically derived by "reversing" the received Routing header unless the integrity and authenticity of the received Source Address and Routing header have been verified (e.g., via the use of an Authentication header in the received packet).

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Packet A

IPv6 Header Source Address: TN2's Global Address Destination Address: NUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 0 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TR1's Global Address
ICMPv6 Echo Request

Procedure:

1. TR1 forwards Packet A, an Echo Request that has a Routing header with three segments. The Echo Request is destined for the NUT.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must transmit an Echo Reply to TN2's global address using TR1 as a first hop. If the Echo Reply contains a Routing header, it must not be a reversal of the received Routing Header.

Possible Problems:

- None.

Test v6LC.1.2.10: Unrecognized Routing Type - End Node

Purpose: Verify that a node properly processes an IPv6 packet destined for it that contains a Routing header with an unrecognized Routing Type value.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If, while processing a received packet, a node encounters a Routing header with an unrecognized Routing Type value, the required behavior of the node depends on the value of the Segments Left field, as follows:

> If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet, whose type is identified by the next header field in the Routing header.
>
> If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Packet A

IPv6 Header Source Address: TN2's Global Address Destination Address: NUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 33 Segments Left: 0 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TR1's Global Address
ICMPv6 Echo Request

Procedure:

1. TR1 forwards Packet A, an Echo Request that has a Routing header with a Routing Type value of 33 and Segments Left value of 0. The Echo Request is destined for the NUT.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must ignore the unrecognized Routing Type value and should respond to the Request by sending an Echo Reply to TN2 using TR1 as the first-hop.

Possible Problems:

- None.

Test v6LC.1.2.11: Unrecognized Routing Type - Intermediate Node (Routers Only)

Purpose: Verify that a router properly processes an IPv6 packet as the intermediate node that contains a Routing header with an unrecognized Routing Type value.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If, while processing a received packet, a node encounters a Routing header with an unrecognized Routing Type value, the required behavior of the node depends on the value of the Segments Left field, as follows:

> If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet, whose type is identified by the next header field in the Routing header.
>
> If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

Test Setup: Common Setup 1.2 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

1. Enable the RUT's interface to Link A.

Packet A

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 33 Segments Left: 1 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Procedure:

1. TN2 transmits Packet A, an Echo Request that has a Routing header with a Routing Type value of 33 and Segments Left value of 1. The Echo Request is destined for TN1 with a first hop through the RUT.
2. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT must discard the Echo Request and send an ICMP Parameter Problem, Code 0, message to TN2's Global Address. The Pointer field must be 0x2A (offset of the Routing Type field of the Routing header).

Possible Problems:

- None.

Test v6LC.1.2.12: Routing Header Reserved Field – End Node

Purpose: Verify that a node properly processes an IPv6 packet destined for it that contains a Routing header with a non-zero Reserved field.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Reserved Field is initialized to zero on transmission and ignored on reception.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Packet A

IPv6 Header Source Address: TN2’s Global Address Destination Address: NUT’s Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 0 Reserved: 0xFFFFFFFF Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TR1’s Global Address
ICMPv6 Echo Request

Procedure:

1. TR1 forwards Packet A to the NUT, an Echo Request that has a Routing header with a non-zero reserved field
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT should ignore the non-zero reserved field and transmit an Echo Reply to TN2’s Global Address using TR1 as a first hop.

Possible Problems:

- None.

Test v6LC.1.2.13: Routing Header Reserved Field – Intermediate Node (Routers Only)

Purpose: Verify that a router properly processes an IPv6 packet destined for it that contains a Routing header with a non-zero Reserved field.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Reserved Field is initialized to zero on transmission and ignored on reception.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

1. Enable the RUT's interface on Link A.

Packet A

IPv6 Header Source Address: TN1's Global Address Destination Address: RUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 3 Reserved: 0xFFFFFFFF Address [1]: TR1's Global Address Address [2]: Global Address 3 Address [3]: TN2's Global Address
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A to the RUT, an Echo Request that has a Routing header with a non-zero reserved field.
2. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT should ignore the non-zero reserved field and forward the Echo Request to TR1's Global Address. The RUT may change the value of the reserved field when the packet is forwarded.

Possible Problems:

- None.

Test v6LC.1.2.14: Routing Header Processing – End Node

Purpose: Verify that a node properly processes an IPv6 packet destined for it that contains a Routing header.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A Routing header is not examined or processed until it reaches the node identified in the Destination Address of the IPv6 header. In that node, dispatching on the Next Header field of the immediately preceding header causes the Routing Header module to be invoked, which, in the case of Routing Type 0, performs the following algorithm:

> If Segments Left = 0, proceed to process the next header in the packet.
>
> Else, if Header Extension Length is odd, send an ICMP Parameter Problem Message, Code 0, to the Source Address pointing to the Header Extension Length field and discard the packet.
>
> Else, compute “n”, the number of addresses in the Routing Header by dividing the Header Extension Length by 2. If Segments Left is greater than n, send an ICMP Parameter Problem Message, Code 0, to the Source Address pointing to the Segments Left field, and discard the packet.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Packet A

IPv6 Header Source Address: TN2’s Global Address Destination Address: NUT’s Global Address Next Header: 43
Routing Header Next Header: 58 **Header Ext. Length: 5** Routing Type: 0 Segments Left: 0 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: First 8 octets of TR1’s Global Address
ICMPv6 Echo Request

Packet B

IPv6 Header Source Address: TN2's Global Address Destination Address: NUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 0 **Address [1]: Solicited-Node Multicast of TR1** Address [2]: Global Address 3 Address [3]: First 8 octets of TR1's Global Address
ICMPv6 Echo Request

Procedure:

Part A: Header Extension Length Odd

1. TR1 transmits Packet A, an Echo Request that has a Routing header with an odd Header Extension Length value of 5. The Echo Request is destined for the NUT.
2. Observe the packets transmitted by the NUT.

Part B: Address[i] is Multicast

3. TN2 transmits Packet B, an Echo Request that has a Routing header with the first address in the Routing Header as TR1's Solicited-Node Multicast Address. The Echo Request is destined for the NUT.
4. Observe the packets transmitted by the NUT.

Observable Results:

- *Parts A, and B*
  Steps 2, and 4: The NUT must ignore the invalid fields of the Routing Header and process the Echo Request message from TN1. The NUT must respond to the Echo Request from TN2 with an Echo Reply.

Possible Problems:

- None.

Test v6LC.1.2.15: Routing Header Processing – Intermediate Node (Routers Only)

Purpose: Verify that a router properly processes an IPv6 packet destined for it that contains a Routing header.

References:

- [IPv6-SPEC] – Sections 4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A Routing header is not examined or processed until it reaches the node identified in the Destination Address of the IPv6 header. In that node, dispatching on the Next Header field of the immediately preceding header causes the Routing Header module to be invoked, which, in the case of Routing Type 0, performs the following algorithm:

> If Segments Left = 0, proceed to process the next header in the packet.
>
> Else, if Header Extension Length is odd, send an ICMP Parameter Problem Message, Code 0, to the Source Address pointing to the Header Extension Length field and discard the packet.
>
> Else, compute “n”, the number of addresses in the Routing Header by dividing the Header Extension Length by 2. If Segments Left is greater than n, send an ICMP Parameter Problem Message, Code 0, to the Source Address pointing to the Segments Left field, and discard the packet.
>
> > Else, decrement Segments Left by 1; compute “i”, the index of the next address to be visited in the address vector, by subtracting Segments Left from “n”. If Address[i] or the IPv6 Destination Address is multicast, discard the packet.
> >
> > > Else, swap the IPv6 Destination Address and Address[i].
> > >
> > > > If the IPv6 Hop Limit is less than or equal to 1, send an ICMP Time Exceeded – Hop Limit Exceeded in Transit message to the Source Address and discard the packet.
> > > >
> > > > Else, decrement the Hop Limit by 1, and resubmit the packet to the IPv6 module for transmission to the new destination.

Test Setup: The Common Setup 1.1 is performed. The Common Cleanup Procedure is performed after each part.

1. Enable the RUT’s interface to Link A.

Packet A

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43
Routing Header Next Header: 58 **Header Ext. Length: 5** Routing Type: 0 Segments Left: 1 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Packet B

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 4 Routing Type: 0 **Segments Left: 3** Address [1]: Global Address 2 Address [2]: TN1's Global Address
ICMPv6 Echo Request

Packet C

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 3 **Address [1]: Solicited-Node Multicast of TR1** Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Packet D

IPv6 Header Source Address: TN2's Global Address **Destination Address: RUT's Solicited-Node Multicast** Next Header: 43
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 3 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Packet E

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43 **Hop Limit: 0**
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 3 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Packet F

IPv6 Header Source Address: TN2's Global Address Destination Address: RUT's Global Address Next Header: 43 **Hop Limit: 1**
Routing Header Next Header: 58 Header Ext. Length: 6 Routing Type: 0 Segments Left: 3 Address [1]: Global Address 2 Address [2]: Global Address 3 Address [3]: TN1's Global Address
ICMPv6 Echo Request

Packet G

IPv6 Header
Source Address: TN2’s Global Address
Destination Address: RUT’s Global Address
Next Header: 43
Hop Limit 10
Routing Header
Next Header: 58
Header Ext. Length: 6
Routing Type: 0
Segments Left: 3
Address [1]: TR1’s Global Address
Address [2]: Global Address 3
Address [3]: TN1’s Global Address
ICMPv6 Echo Request

Procedure:

Part A: Header Extension Length Odd

1. TN2 transmits Packet A, an Echo Request that has a Routing header with an odd Header Extension Length value of 5. The Echo Request is destined for TN1 with a first hop through the RUT.
2. Observe the packets transmitted by the RUT.

Part B: Segments Left Greater Than Number of Addresses

3. TN2 transmits Packet B, an Echo Request that has a Routing header with a Segments Left field that is greater than the number of addresses. The Echo Request is destined for TN1 with a first hop through the RUT.
4. Observe the packets transmitted by the RUT.

Part C: Address[i] is Multicast

5. TN2 transmits Packet C, an Echo Request that has a Routing header with the first address in the Routing Header as TR1’s Solicited-Node Multicast Address. The Echo Request is destined for TN1 with a first hop through the RUT.
6. Observe the packets transmitted by the RUT.

Part D: Destination is Multicast

7. TN2 transmits Packet D, an Echo Request that has a Routing header with an IPv6 destination of the RUT’s Solicited-Node Multicast Address. The Echo Request is destined for TN1 with a first hop through the RUT.
8. Observe the packets transmitted by the RUT.

Part E: Hop Limit = 0.

9. TN2 transmits Packet E, an Echo Request that has a Routing header with a Hop Limit field of value 0 in the IPv6 header. The Echo Request is destined for TN1 with a first hop through the RUT.
10. Observe the packets transmitted by the RUT.

Part F: Hop Limit = 1.

11. TN2 transmits Packet F, an Echo Request that has a Routing header with a Hop Limit field of

value 1 in the IPv6 header. The Echo Request is destined for TN1 with a first hop through the RUT.

12. Observe the packets transmitted by the RUT.

Part G: Valid Processing

13. TN2 transmits Packet G, an Echo Request that has a Routing header with three segments, the first of which is through TR1. The Echo Request is destined for TN1 with a first hop through the RUT.
14. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*
  Step 2: The RUT must discard the Echo Request and send an ICMPv6 Parameter Problem, Code 0 (hop limit exceeded in transit), message to TN2's Global Address. The Pointer field must be 0x29 (offset of the Header Extension Length field of the Routing header).
- *Part B*
  Step 4: The RUT must discard the Echo Request and send an ICMPv6 Parameter Problem, Code 0 (hop limit exceeded in transit), message to TN2's Global Address. The Pointer field must be 0x2B (offset of the Segments Left field of the Routing header).
- *Part C*
  Step 6: The RUT must discard the Echo Request.
- *Part D*
  Step 8: The RUT must discard the Echo Request.
- *Part E*
  Step 10: The RUT must discard the Echo Request and transmit an ICMPv6 Time Exceeded - Hop Limit Exceeded in Transit Message to TN2's Global Address.
- *Part F*
  Step 12: The RUT must discard the Echo Request and transmit an ICMPv6 Time Exceeded - Hop Limit Exceeded in Transit Message to TN2's Global Address.
- *Part G*
  Step 14: The RUT must transmit the Echo Request to TR1. The packet must have the following values:

IPv6 Source Address	TN2's Global Address
IPv6 Destination Address	TR1's Global Address
Hop Limit	9
Routing Header:	
Hdr. Ext. Length	6
Routing Type	0
Segments Left	2
Address[1]	RUT's Global Address
Address[2]	Global Address 3
Address[3]	TN1's Global Address

Possible Problems:

- None.

Group 3: Fragmentation

Scope

The following tests cover fragmentation in IPv6.

Overview

The tests in this group verify that a node properly times out fragment reassembly, abandons reassembly on packets that exceed a maximum size, processes stub fragments, and reassembles overlapping fragments. These tests also verify that a node generates the proper ICMPv6 messages.

Test v6LC.1.3.1: Fragment Reassembly

Purpose: Verify that a node correctly reassembles fragmented packets and distinguishes between packet fragments using the Source Address, Destination Address, and Fragment ID.

References:

- [IPv6-SPEC] – Sections 4.5 and 5
- [ICMPv6] – Section 3.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Fragment header is used by an IPv6 source to send a packet larger than would fit in the Path MTU to its destination. An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address and Fragment Identification. If insufficient fragments are received to complete reassembly of a packet within 60 seconds of the reception of the first-arriving fragment of that packet, reassembly of that packet must be abandoned and all the fragments that have been received for that packet must be discarded. If the first fragment (i.e., the one with a Fragment Offset of zero) has been received, an ICMP Time Exceeded -- Fragment Reassembly Time Exceeded message should be sent to the source of that fragment.

A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 packet with an ICMPv6 Echo Request with 80 bytes of data fragmented into three packets, the maximum of which contains 32 bytes of payload.

Fragment A.1	Fragment A.2	Fragment A.3
IPv6 Header Next Header: 44 Source Address: [See below] Destination Address: [See below]	IPv6 Header Next Header: 44 Source Address: [See below] Destination Address: [See below]	IPv6 Header Next Header: 44 Source Address: [See below] Destination Address: [See below]
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1 ID: [See below]	Fragment Header Next Header: 58 Fragment Offset: (4) 32 bytes More Fragments flag: 1 ID: [See below] Fragment Data: 32 Bytes	Fragment Header Next Header: 58 Fragment Offset: (8) 64 bytes More Fragments flag: 0 ID: [See below] Fragment Data: 24 Bytes
ICMPv6 Echo Request		

Procedure:

Part A: All Fragments are Valid

1. TN1 transmits Fragments A.1, A.2, and A.3 in order. All fragments have the same Source Address, Destination Address, and Fragment ID.
2. Observe the packets transmitted by the NUT.

Part B: All Fragments are Valid, reverse order

3. TN1 transmits Fragments A.3, A.2, and A.1, in that order. All fragments have the same Source Address, Destination Address, and Fragment ID.
4. Observe the packets transmitted by the NUT.

Part C: Fragment IDs Differ Between Fragments

5. TN1 transmits Fragments A.1, A.2, and A.3 in order. Fragments A.1 and A.3 have a Fragment ID of 2999. Fragment A.2 has a Fragment ID of 3000. The Source and Destination Addresses for all fragments are the same.
6. Observe the packets transmitted by the NUT.

Part D: Source Addresses Differ Between Fragments

7. TN1 transmits Fragments A.1, A.2, and A.3 in order. Fragments A.1 and A.3 have a Source Address of the link-local address of TN1. Fragment A.2 has a Source Address of a different link-local address. The Destination Addresses and Fragment Ids for all fragments are the same.
8. Observe the packets transmitted by the NUT.

Part E: Destination Address Differ Between Fragments

9. TN1 transmits Fragments A.1, A.2, and A.3 in order. Fragments A.1 and A.3 have a Destination Address of the link-local address of the NUT. Fragment A.2 has a Destination Address of the global address of the NUT. The Source Addresses and Fragment Ids for all fragments are the same.
10. Observe the packets transmitted by the NUT.

Part F: Reassemble to 1500

11. TN1 transmits an Echo Request to the NUT. TN1 answers any Neighbor Solicitation with a Neighbor Advertisement.
12. Observe the packets transmitted by the NUT.
13. TN1 transmits Fragments A.1, A.2, and A.3 in order. All fragments have the same Source Address, Destination Address, and Fragment ID, however, the payloads of each fragment are modified so that the reassembled packet size is 1500.
14. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must transmit an Echo Reply to TN1 in response to the reassembled Echo Request.
- *Part B*

  Step 4: The NUT must transmit an Echo Reply to TN1 in response to the reassembled Echo Request.
- *Part C*

Step 6: The NUT must not transmit an Echo Reply to TN1, as the Echo Request could not be reassembled due to differences in the Fragment ID. The NUT should transmit an ICMPv6 Time Exceeded Message to TN1 sixty seconds after reception of Fragment A.1.

- *Part D*

  Step 8: The NUT must not transmit an Echo Reply to TN1, as the Echo Request could not be reassembled due to differences in the Source Address. The NUT should transmit an ICMPv6 Time Exceeded Message to TN1 sixty seconds after reception of Fragment A.1.
- *Part E*

  Step 10: The NUT must not transmit an Echo Reply to TN1, as the Echo Request could not be reassembled due to differences in the Destination Address. The NUT should transmit an ICMPv6 Time Exceeded Message to TN1 sixty seconds after reception of Fragment A.1.
- *Part F*

  Step 12: The NUT must respond to the Echo Request from TN1.
  Step 14: The NUT must respond to the Echo Request from TN1.

Possible Problems:

- None.

Test v6LC.1.3.2: Reassembly Time Exceeded

Purpose: Verify that a node takes the proper actions when the reassembly time has been exceeded for a packet.

References:

- [IPv6-SPEC] – Section 4.5
- [ICMPv6] –Section 2.2, 3.3, 2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If insufficient fragments are received to complete reassembly of a packet within 60 seconds of the reception of the first-arriving fragment of the packet, reassembly of that packet must be abandoned. Further, all the fragments that have been received for that packet must be discarded. If the first fragment (i.e., the one with a Fragment Offset of zero) has been received, an ICMPv6 Time Exceeded – Fragment Reassembly Time Exceeded message should be sent to the source of that fragment.

The unused field MUST be initialized to zero by the sender and ignored by the receiver.

If the message is a response to a message sent to one of the node’s unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Common Test Setup 1.1 is performed before each part. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 packet with an ICMPv6 Echo Request with 80 bytes of data fragmented into three packets, the maximum of which contains 32 bytes of payload.

Fragment A.1	Fragment A.2	Fragment A.3
IPv6 Header Next Header: 44 Source Address: TN1's Global Address Destination Address: NUT's Global Address	IPv6 Header Next Header: 44 Source Address: TN1's Global Address Destination Address: NUT's Global Address	IPv6 Header Next Header: 44 Source Address: TN1's Global Address Destination Address: NUT's Global Address
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1	Fragment Header Next Header: 58 Fragment Offset: (4) 32 bytes More Fragments flag: 1 Fragment Data: 32 Bytes	Fragment Header Next Header: 58 Fragment Offset: (8) 64 bytes More Fragments flag: 0 Fragment Data: 24 Bytes
ICMPv6 Echo Request		

Procedure:

Part A: Time Elapsed Between Fragments less than Sixty Seconds

1. TN1 transmits Fragments A.1, A.2 and A.3 in order. There is a 55-second delay between the transmission of Fragment A.1 and Fragments A.2 and A.3.
2. Observe the packets transmitted by the NUT.

Part B: Time Exceeded Before Last Fragments Arrive

3. TN1 transmits Fragments A.1, A.2 and A.3 in order. There is a 65-second delay between the transmission of Fragment A.1 and Fragments A.2 and A.3.
4. Observe the packets transmitted by the NUT.

Part C: Time Exceeded, Only First Fragment Received

5. TN1 transmits Fragment A.1
6. Observe the packets transmitted by the NUT.

Part D: Time Exceeded, Only Second Fragment Received

7. TN1 transmits Fragment A.2.
8. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: Fragments A.2 and A.3 arrive just before the NUT's reassembly timer expires for Fragment A.1. The NUT must transmit an Echo Reply to TN1 in response to the reassembled Echo Request.
- *Part B*

  Step 4: Fragments A.2 and A.3 arrive after the NUT's reassembly timer expires for Fragment A.1. The NUT must not transmit an Echo Reply to TN1, as the Echo Request could not be reassembled in time. The NUT should transmit an ICMPv6 Time Exceeded Message to TN1 sixty seconds after reception of Fragment A.1 with a code field value of 1 (Fragment Reassembly Time Exceeded).

    - The unused field must be initialized to zero.
    - The Source Address of the Packet must be the same as the Global Destination Address of TN1’s Echo Request packet.
    - The Destination Address should be the same as the Global Source Address of TN1’s Echo Request packet.
    - The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

- *Part C*

  Step 6: The NUT must not transmit an Echo Reply to TN1, as the Echo Request was not completed. The NUT should transmit an ICMPv6 Time Exceeded Message to TN1 sixty seconds after reception of Fragment A.1 with a code field value of 1 (Fragment Reassembly Time Exceeded).

    - The unused field must be initialized to zero.
    - The Source Address of the Packet must be the same as the Global Destination Address of TN1’s Echo Request packet.
    - The Destination Address should be the same as the Global Source Address of TN1’s Echo Request packet.
    - The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

- *Part D*

  Step 8: The NUT must not transmit an Echo Reply or a Time Exceeded Message to TN1.

Possible Problems:

- None.

Test v6LC.1.3.3: Fragment Header M-Bit Set, Payload Length Invalid

Purpose: Verify that a node takes the proper actions when it receives a fragment with the M-bit set (more fragments), but which has a Payload Length that is not a multiple of 8 bytes.

References:

- [IPv6-SPEC] – Section 4.5
- [ICMPv6] – Section 3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If the length of a fragment, as derived from the fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is set, then that fragment must be discarded. Also, an ICMPv6 Parameter Problem message should be sent to the packet's source with an ICMPv6 Code field of 0 and an ICMPv6 Pointer field containing the offset of the Payload Length field of the fragment packet.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A
IPv6 Header Payload Length: 21 bytes Next Header: 44
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1
ICMPv6 Echo Request Data Length: 5 bytes

Procedure:

1. TN1 transmits Packet A, an Echo Request that has a Fragment header with the M-bit set. The Payload Length is 21, which is not a multiple of 8 octets.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must not transmit an Echo Reply to TN1, as the fragment was discarded. The NUT should transmit an ICMPv6 Parameter Problem message to TN1. The Code field should be 0 (erroneous header field encountered). The Pointer field should be 0x04 (offset of Payload Length field of the IPv6 header).

Possible Problems:

- None.

Test v6LC.1.3.4: Stub Fragment Header

Purpose: Verify that a node accepts the offset zero fragment with the More Fragments flag clear.

References:

- [IPv6-SPEC] – Sections 4.5, and 5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Fragment Offset of the first fragment is 0. An M flag value of 0 indicates the fragment is the last fragment. It is acceptable for a fragmented packet to consist of only a single fragment.

In response to an IPv6 packet that is sent to an IPv4 destination (i.e. a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big Message reporting a Next-Hop MTU of less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment Header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments.

Test Setup: No Common Test Setup is performed. The Common Test Cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 44
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 0
ICMPv6 Echo Request

Procedure:

1. TN1 transmits Packet A, an Echo Request that has a Fragment header with a Fragment Offset of 0 and the More Fragments flag clear.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must transmit an Echo Reply to TN1. The Echo Reply must not include a Fragment header.

Possible Problems:

- None.

Section 2: RFC 2461

Scope

The following tests cover the Neighbor Discovery Specification for Internet Protocol version 6, Request For Comments 2461. The Neighbor Discovery protocol is used by nodes to determine the link-layer address for neighbors known to reside on attached links as well as to quickly purge cached values that become invalid. Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to actively keep track of neighbors that are reachable and those that are not. When a router or the path to a router fails, a host actively searches for functioning alternates.

Overview

These tests are designed to verify the readiness of an IPv6 implementation vis-à-vis the Neighbor Discovery specification.

Default Packets

Echo Request

IPv6 Header Next Header: 58
ICMPv6 Header Type: 128 Code: 0

*Note: Due to the nature of the STALE state, one cannot verify state STALE without causing the state itself to change. For this reason, in tests where we require the NCE to transition from STALE to another state (except DELAY), we cannot verify state STALE with an observable action.

Router Advertisement

IPv6 Header Source Address: TR1's Link-Local Address Destination Address: All-Nodes multicast address Next Header: 58
ICMPv6 Header Type: 134 Code: 0 M Bit (managed): 0 O Bit (other): 0 Router Lifetime: 20 seconds Reachable Time: 10 seconds Retrans Timer: 1 second
Prefix Option Type: 3 L Bit (on-link flag): 1 A Bit (addr conf): 1 Valid Lifetime: 20 seconds Preferred Lifetime: 20 seconds

Redirect message

IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address Destination Address: NUT's Link Local Address
ICMPv6 Header Type: 137 Code: 0
Redirected Header Option Type: 4 Length: Length of Invoking Packet in 8 octet units
Invoking Packet

Group 1: Address Resolution and Neighbor Unreachability Detection

Scope

The following tests cover Address Resolution and Neighbor Unreachability Detection in IPv6.

Overview

The tests in this group verify conformance of the Address Resolution and Neighbor Unreachability Detection function with the Neighbor Discovery Specification.

Test v6LC.2.1.1: On-link Determination

Purpose: Verify that a node correctly determines that a destination is on-link.

References:

- [ND] – Section 5.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Next-hop determination for a given unicast destination operates as follows. The sender performs a longest prefix match against the Prefix List to determine whether the packet's destination is on- or off-link. If the destination is on-link, the next-hop address is the same as the packet's Destination Address. Otherwise, the sender selects a router from the Default Router List. If the Default Router List is empty, the sender assumes that the destination is on-link.

Test Setup: No Common Test Setup is performed. If the NUT is a router, configure a default route with TR1 as next hop. Part B does not require a default route. The common cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address	IPv6 Header Next Header: 58 Source Address: TN1's Global Address
ICMPv6 Echo Request	ICMPv6 Echo Request

Router Advertisement
IPv6 Header Next Header: 58
Router Advertisement Prefix Length: 64 L Bit: 1 (on-link) Prefix: TN1's Global Prefix

Packet C
IPv6 Header Next Header: 58 Source Address: TN2's Global Address Destination Address: NUT's Global Address
ICMPv6 Echo Request

Procedure:

Part A: Link-local Address

1. TN1 transmits Packet A an Echo Request with TN1's link-local source address.
2. Observe the packets transmitted by the NUT.

Part B: Global Address, No Default Router

3. TN1 transmits Packet B, an Echo Request with TN1's global source address.
4. Observe the packets transmitted by the NUT.

Part C: Global Address, On-link Prefix covers TN1

5. TR1 transmits the Router Advertisement. The Prefix Advertisement covers TN1's global address.
6. TN1 transmits Packet B, an Echo Request with TN1's global source address.
7. Observe the packets transmitted by the NUT.

Part D: Global Address, On-link Prefix does not cover TN2

8. TR1 transmits the Router Advertisement. The Prefix Advertisement does not cover TN2's global address.
9. TN2 transmits Packet C, an Echo Request with TN2's global source address.
10. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT should send a Neighbor Solicitation with Target Address equal to TN1's link-local address, indicating that the NUT has successfully determined that TN1 was on-link.
- *Part B*

  Step 4: There are no routers on the link, thus the NUT must not reply to a default router or solicit for its default router.
- *Part C*

  Step 7: TN1's global address is covered by the on-link prefix. Hence, the NUT should consider TN1's global address as on-link. The NUT should send a Neighbor Solicitation with Target Address equal to TN1's global address, indicating that the NUT has successfully determined that TN1 was on-link.
- *Part D*

  Step 10: TN2's global address is not covered by the on-link prefix. Hence, the NUT should consider TN2's global address as off-link. The NUT should send a Neighbor Solicitation with Target Address equal to TR1's link-local address indicating that the NUT has successfully determined that TN2 was off-link.

Possible Problems:

- None.

Test v6LC.2.1.2: Resolution Wait Queue

Purpose: Verify that a node properly queues packets while waiting for address resolution of the next hop.

References:

- [ND] – Section 3, Section 7.2.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Nodes accomplish address resolution by multicasting a Neighbor Solicitation that asks the target node to return its link-layer address. Neighbor Solicitation messages are multicast to the solicited-node multicast address of the target address. The target returns its link-layer address in a unicast Neighbor Advertisement message. A single request-response pair of packets is sufficient for both the initiator and the target to resolve each other's link-layer addresses; the initiator includes its link-layer address in the Neighbor Solicitation.

While waiting for address resolution to complete, the sender MUST, for each neighbor, retain a small queue of packets waiting for address resolution to complete. The queue MUST hold at least one packet, and MAY contain more. However, the number of queued packets per neighbor SHOULD be limited to some small value. When a queue overflows, the new arrival SHOULD replace the oldest entry. Once address resolution completes, the node transmits any queued packets.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Packet A

Ipv6 Header Next Header: 58 Source Address: TN1's Link-local Address
ICMPv6 Echo Request ID: 3

Packet B

Ipv6 Header Next Header: 58 Source Address: TN2's Link-local Address
ICMPv6 Echo Request ID: 4

Neighbor Advertisement C

IPv6 Header Next Header: 58 Source Address: TN1’s Link-local Address Destination Address: NUT’s Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1 Target Address: TN1’s Link-local Address

Neighbor Advertisement D

IPv6 Header Next Header: 58 Source Address: TN2’s Link-local Address Destination Address: NUT’s Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1 Target Address: TN2’s Link-local Address

Procedure:

Part A: Single Queue

1. TN1 transmits Packet A, an Echo Request, 3 times. The ID is incremented each time.
2. Observe the packets transmitted by the NUT.
3. TN1 transmits the Neighbor Advertisement C in response to any Neighbor Solicitations from the NUT.
4. Observe the packets transmitted by the NUT.

Part B: Multiple Queues

5. TN1 transmits Packet A, an Echo Request, 3 times. The ID is incremented each time.
6. TN2 transmits Packet B, an Echo Request, 4 times. The ID is incremented each time.
7. Observe the packets transmitted by the NUT.
8. TN1 and TN2 transmit the Neighbor Advertisement C and D respectively in response to any Neighbor Solicitations from the NUT.
9. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*
  Step 2: The NUT should transmit a Neighbor Solicitation with a Target Address equal to TN1’s link-local address. The NUT should send Echo Replies to TN1 in response to Packet A.
  Step 4: The Echo Replies should correspond to the last 3 Echo Requests sent by TN1 to the NUT, indicating successful queuing of packets while waiting for address resolution to complete. The number of Echo Replies MUST be no less than 1.
- *Part B*
  Step 6: The NUT should transmit a Neighbor Solicitation with a Target Address equal to TN1’s link-local address. The NUT should send Echo Replies to TN1 in response to Packet A. The NUT should transmit a Neighbor Solicitation with a Target Address equal to TN2’s link-local address. The NUT should send Echo Replies to TN2 in response to Packet B.
  Step 9: The Echo Replies should correspond to the last 3 Echo Requests sent by TN1 to the NUT, indicating successful queuing of packets while waiting for address resolution to complete. The number of Echo Replies MUST be no less than 1. The Echo Replies should correspond to the last 4 Echo Requests sent by TN2 to the NUT, indicating successful queuing of packets while waiting for address resolution to complete. The number of Echo Replies MUST be no less than 1.

Possible Problems:

- None.

Test v6LC.2.1.3: Prefix Information Option Processing, On-link Flag (Hosts Only)

Purpose: Verify that a host properly processes the on-link flag of a Prefix Information Option.

References:

- [ND] – Section 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Prefix Information options that have the "on-link" (L) flag set indicate a prefix identifying a range of addresses that should be considered on-link. Note, however, that a Prefix Information option with the on-link flag clear conveys no information concerning on-link determination and MUST NOT be interpreted to mean that addresses covered by the prefix are off-link.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Router Advertisement A

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: All-nodes Multicast Address
Router Advertisement Router Lifetime: 100 seconds Reachable Time: 10 seconds Retransmit Interval: 1 second
Prefix Option "on-link" (L) flag: 1 Valid Lifetime: 20 seconds Preferred Lifetime: 20 seconds Prefix: TR1's Global Prefix

Packet A

IPv6 Header Next Header: 58 Source Address: TR1's Global Address Destination Address: HUT's Link-local Address
ICMPv6 Echo Request

Procedure:

1. TR1 transmits Router Advertisement A.
2. TR1 transmits Packet A. TR1 should not respond to Neighbor Solicitations from the HUT.
3. Observe the packets transmitted by the HUT.
4. TR1 transmits Router Advertisement A with the on-link (L) flag clear.
5. TR1 transmits Packet A. TR1 should not respond to Neighbor Solicitations from the HUT.
6. Observe the packets transmitted by the HUT.

Observable Results:

Step 3: In response to Packet A, the HUT should transmit 3 Neighbor Solicitations with a Target Address of TR1's global address.
Step 6: In response to Packet A, the HUT should transmit 3 Neighbor Solicitations with a Target Address of TR1's global address.

Possible Problems:

- None.

Test v6LC.2.1.4: Host Prefix List (Hosts Only)

Purpose: Verify that a host properly updates its Prefix List upon receipt of Prefix Information Options, which have the on-link flag set.

References:

- [ND] – Sections 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: For each Prefix Information option with the on-link flag set, a host does the following:

- If the prefix is not already present in the Prefix List, and the Prefix Information option's Valid Lifetime field is non-zero, create a new entry for the prefix and initialize its invalidation timer to the Valid Lifetime value in the Prefix Information option.
- If the prefix is already present in the host's Prefix List as the result of a previously received advertisement, reset its invalidation timer to the Valid Lifetime value in the Prefix Information option. If the new Lifetime value is zero, timeout the prefix immediately (see Section 6.3.5).
- If the Prefix Information option's Valid Lifetime field is zero, and the prefix is not present in the host's Prefix List, silently ignore the option.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Advertisement A

IPv6 Header Next Header: 58 Source Address: TR1’s Link-local Address Destination Address: All-nodes Multicast Address
Router Advertisement Router Lifetime: 20 seconds Reachable Time: 600 seconds Retransmit Interval: 1 second
Prefix Option “on-link” (L) flag: 1 Valid Lifetime: 10 seconds Preferred Lifetime: 10 seconds Prefix: TN1’s Global Prefix

Packet B

IPv6 Header Next Header: 58 Source Address: TN1's Global Address Destination Address: HUT's Link-local Address
ICMPv6 Echo Request

Procedure:

Part A: Prefix Lifetime has not Expired

1. TR1 transmits Router Advertisement A without the Prefix Option.
2. TR1 transmits a link-local Echo Request to the HUT.
3. Observe the packets transmitted by the HUT. TR1 transmits a Neighbor Advertisement in response to any Neighbor Solicitations from the HUT.
4. TR1 transmits Router Advertisement A. The Source Address is the TR1's Link-local Address. The Destination Address is the multicast address. The on-link flag is set. Wait 8 seconds.
5. TN1 transmits Packet B, whose Source Address is covered by the prefix advertised in Router Advertisement A.
6. Observe the packets transmitted by the HUT.

Part B: Prefix Lifetime updated by Router Advertisement

7. TR1 transmits Router Advertisement A without the Prefix Option.
8. TR1 transmits a link-local Echo Request to the HUT.
9. Observe the packets transmitted by the HUT. TR1 transmits a Neighbor Advertisement in response to any Neighbor Solicitations from the HUT.
10. TR1 transmits Router Advertisement A. Wait 8 seconds.
11. TR1 transmits Router Advertisement A. Wait 8 seconds.
12. TN1 transmits Packet B, whose Source Address is covered by the prefix advertised in Router Advertisement A.
13. Observe the packets transmitted by the HUT.

Observable Results:

- *Part A*

  Step 3: The HUT should solicit and reply to the Echo Request transmitted by TR1.
  Step 6: In response to Packet B, the HUT should transmit Neighbor Solicitations with a Target Address of TN1's global address.
- *Part B*

  Step 9: The HUT should solicit and reply to the Echo Request transmitted by TR1.
  Step 13: In response to Packet B, the HUT should transmit Neighbor Solicitations with a Target Address of TN1's global address.

Possible Problems:

- None.

Test v6LC.2.1.5: Neighbor Solicitation Origination, Address Resolution

Purpose: Verify that a node properly originates Neighbor Solicitations when trying to resolve the address of a neighbor.

References:

- [ND] – Sections 6.2.1, 7.2.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a node has a unicast packet to send to a neighbor, but does not know the neighbor's link-layer address, it performs address resolution. For multicast-capable interfaces this entails creating a Neighbor Cache entry in the INCOMPLETE state and transmitting a Neighbor Solicitation message targeted at the neighbor. The solicitation is sent to the solicited-node multicast address corresponding to the target address.

If the source address of the packet prompting the solicitation is the same as one of the addresses assigned to the outgoing interface, that address SHOULD be placed in the IP Source Address of the outgoing solicitation. Otherwise, any one of the addresses assigned to the interface should be used. Using the prompting packet's source address when possible insures that the recipient of the Neighbor Solicitation installs in its Neighbor Cache the IP address that is highly likely to be used in subsequent return traffic belonging to the prompting packet's "connection". If the solicitation is being sent to a solicited-node multicast address, the sender MUST include its link-layer address (if it has one) as a Source Link-Layer Address option. Including the source link-layer address in a multicast solicitation is required to give the target an address to which it can send the Neighbor Advertisement. On unicast solicitations, an implementation MAY omit the Source Link-Layer Address option. The assumption here is that if the sender has a peer's link-layer address in its cache, there is a high probability that the peer will also have an entry in its cache for the sender. Consequently, it need not be sent.

While awaiting a response, the sender SHOULD retransmit Neighbor Solicitation messages approximately every RetransTimer milliseconds, even in the absence of additional traffic to the neighbor. Retransmissions MUST be rate-limited to at most one solicitation per neighbor every RetransTimer milliseconds.

Some host variables (e.g., CurHopLimit, RetransTimer, and ReachableTime) apply to all nodes including routers. In practice, these variables may not actually be present in routers, since their contents can be derived from the router configuration variables. However, external router behavior MUST be the same as host behavior with respect to these variables.

Test Setup: The common cleanup procedure is performed after each part.

Packet A	Packet B
IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address	IPv6 Header Next Header: 58 Source Address: TN1's Global Address Destination Address: NUT's Global Address
ICMPv6 Echo Request	ICMPv6 Echo Request

Procedure:

Part A: Neighbor Solicitation Origination, Target Address Being Link-local

1. If the NUT is a host, perform Common Test Setup 1.1 with a Retransmit Interval value of 1 second. If the NUT is a router, configure the Retransmit Interval value to 1 second.
2. TN1 transmits Packet A. The source address is TN1's link-local address and the destination address is the NUT's link-local address.
3. Observe the packets transmitted by the NUT. TN1 doesn't send any Neighbor Advertisement.
4. Repeat Steps 1 and 2 with a Retransmit Interval value of 5 seconds and observe the packets transmitted by the NUT.

Part B: Neighbor Solicitation Origination, Target Address Being Global

5. If the NUT is a host, perform Common Test Setup 1.1 with a Retransmit Interval value of 1 second. If the NUT is a router, configure the Retransmit Interval value to 1 second.
6. TN1 transmits Packet B. The source address is TN1's global address and the destination is the NUT's global address.
7. Observe the packets transmitted by the NUT. TN1 doesn't send any Neighbor Advertisement.
8. Repeat Steps 5 and 6 with a Retransmit Interval value of 5 seconds and observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 3: In response to Packet A, the NUT should transmit Neighbor Solicitations with a Target Address equal to the TN1's Link-local Address at intervals of 1 second. The NUT MUST transmit no more than 1 Neighbor Solicitation every 1 second. Each Neighbor Solicitation MUST have a Source Link-Layer Address Option. The maximum number of Neighbor Solicitations should be MAX_MULTICAST_SOLICIT, which should be 3.
  Step 4: The NUT should transmit the Neighbor Solicitations with a Target Address equal to the TN1's Link-local Address at intervals of 5 seconds. The NUT MUST transmit no more than 1 Neighbor Solicitation every 5 seconds. Each Neighbor Solicitation MUST have a Source Link-Layer Address Option. The maximum number of Neighbor Solicitations should be MAX_MULTICAST_SOLICIT, which should be 3.
- *Part B*

Step 7: In response to Packet B, the NUT should transmit Neighbor Solicitations with a Target Address equal to the TN1's Global Address at intervals of 1 second. The NUT MUST transmit no more than 1 Neighbor Solicitation every 1 second. Each Neighbor Solicitation MUST have a Source Link-Layer Address Option. The maximum number of Neighbor Solicitations should be MAX_MULTICAST_SOLICIT, which should be 3.
Step 8: The NUT should transmit the Neighbor Solicitations with a Target Address equal to the TN1's Global Address at intervals of 5 seconds. The NUT MUST transmit no more than 1 Neighbor Solicitation every 5 seconds. Each Neighbor Solicitation MUST have a Source Link-Layer Address Option. The maximum number of Neighbor Solicitations should be MAX_MULTICAST_SOLICIT, which should be 3.

Possible Problems:

- None.

Test v6LC.2.1.6: Neighbor Solicitation Origination, Reachability Confirmation

Purpose: Verify that a node properly originates Neighbor Solicitations when trying to confirm the reachability of a neighbor.

References:

- [ND] – Section 7.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A neighbor is considered reachable if the node has recently received a confirmation that packets sent recently to the neighbor were received by its IP layer. Positive confirmation can be gathered in two ways: hints from upper layer protocols that indicate a connection is making "forward progress", or receipt of a Neighbor Advertisement message that is a response to a Neighbor Solicitation message.

A connection makes "forward progress" if the packets received from a remote peer can only be arriving if recent packets sent to that peer are actually reaching it. In TCP, for example, receipt of a (new) acknowledgement indicates that previously sent data reached the peer. Likewise, the arrival of new (non-duplicate) data indicates that earlier acknowledgements are being delivered to the remote peer. If packets are reaching the peer, they must also be reaching the sender's next-hop neighbor; thus "forward progress" is a confirmation that the next-hop neighbor is reachable. For off-link destinations, forward progress implies that the first-hop router is reachable. When available, this upper-layer information SHOULD be used.

In some cases (e.g., UDP-based protocols and routers forwarding packets to hosts) such reachability information may not be readily available from upper-layer protocols. When no hints are available and a node is sending packets to a neighbor, the node actively probes the neighbor using unicast Neighbor Solicitation messages to verify that the forward path is still working.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

The Reachable Time is 30 seconds in the Router Advertisement transmitted by TR1 in Common Test Setup 1.1.

Packet A

IPv6 Header Next Header: 58 Source Address: TN1’s Link-local Address Destination Address: NUT’s Link-local Address
ICMPv6 Echo Request

Procedure:

1. Perform Common Test Setup 1.1 with a Retransmit Interval value of 1 second.
2. TN1 transmit Packet A. The source address is TN1’s link-local address and the destination address is the NUT’s link-local address.
3. Observe the packets transmitted by the NUT. TN1 sends a Neighbor Advertisement upon receiving Neighbor Solicitations from the NUT.
4. Wait REACHABLE_TIME * MAX_RANDOM_FACTOR seconds so that the NCE of TN1 transit to state STALE.
5. TN1 transmits Packet A. The source address is TN1’s Link-local address and the destination address is the NUT’s Link-local address.
6. Observe the packets transmitted by the NUT.
7. Wait DELAY_FIRST_PROBE_TIME seconds so that NCE of TN1 transit to state PROBE.
8. Observe the packets transmitted by the NUT.

Observable Results:

Step 3: In response to Packet A, the NUT should transmit Neighbor Solicitations with a Target Address equal to the TN1’s link-local Address at intervals of 1 second. The NUT MUST transmit no more than 1 Neighbor Solicitation every 1 second. Once a Neighbor Advertisement is received from TN1, the NUT should send an Echo Reply in response to Packet A. The NCE of TN1 is in state REACHABLE.
Step 6: In response to Packet A, the NUT should transmit an Echo Reply.
Step 8: The NUT should transmit Neighbor Solicitations with the NUT’s link-local address being the source address and TN1’s link-local address as the destination address. The maximum number of Neighbor Solicitations that the NUT can transmit is 3.

Possible Problems:

- None.

Test v6LC.2.1.7: Invalid Neighbor Solicitation Handling

Purpose: Verify that a node takes the proper actions upon receipt of an invalid Neighbor Solicitation.

References:

- [ND] – Sections 7.1.1 and 7.2.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node MUST silently discard any received Neighbor Solicitation messages that do not satisfy all of the following validity checks:

- The IP Hop Limit field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.

- If the message includes an IP Authentication Header, the message authenticates correctly.

- ICMP Checksum is valid.

- ICMP Code is 0.

- ICMP length (derived from the IP length) is 24 or more octets.

- Target Address is not a multicast address.

- All included options have a length that is greater than zero.

- If the IP source address is the unspecified address, the IP destination address is a solicited-node multicast address.

- If the IP source address is the unspecified address, there is no source link-layer address option in the message.

A valid Neighbor Solicitation that does not meet any the following requirements MUST be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],

- The Target Address is a unicast address for which the node is offering proxy service, or

- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

4.3. Neighbor Solicitation Message Format

IP Fields:
Possible options:
  Source link-layer address
    The link-layer address for the sender. On link layers that have addresses this option MUST be included in multicast solicitations and SHOULD be included in unicast solicitations.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Neighbor Sol. A	Neighbor Sol. B	Neighbor Sol. C
IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address Hop Limit: 255	IPv6 Header Next Header: 58 Source Address: Unspecified Address Destination Address: NUT's Link-local Address Hop Limit: 255	IPv6 Header Next Header: 58 Source Address: Unspecified Address Destination Address: NUT's Solicited-node Multicast Address Hop Limit: 255
Neighbor Sol. Target Address: NUT's Link-local Address Souce Link-layer Address: TN1's Link-layer address	Neighbor Sol. Target Address: NUT's Link-local Address	Neighbor Sol. Target Address: NUT's Link-local Address Source Link-layer Address: TN1's Link-layer address

Procedure:

Part A: Invalid Target Address
1. TN1 transmits Neighbor Solicitation A with the Target Address set to the All Nodes Multicast.
2. Observe the packets transmitted by the NUT.

Part B: Invalid Destination Address
3. TN1 transmits Neighbor Solicitation B.
4. Observe the packets transmitted by the NUT.

Part C: Invalid Source Link-layer Address Option
5. TN1 transmits Neighbor Solicitation C.
6. Observe the packets transmitted by the NUT.

Part D: Invalid Hop Limit
7. TN1 transmits Neighbor Solicitation A with the Hop Limit set to 254.
8. Observe the packets transmitted by the NUT.

Part E: Invalid Checksum

9. TN1 transmits Neighbor Solicitation A with the ICMP checksum set to be invalid.
10. Observe the packets transmitted by the NUT.

Part F: Invalid ICMP code

11. TN1 transmits Neighbor Solicitation A with the ICMP Code set to 1.
12. Observe the packets transmitted by the NUT.

Part G: Invalid ICMP Length

13. TN1 transmits Neighbor Solicitation A with the ICMP Length set to 16.
14. Observe the packets transmitted by the NUT.

Part H: Option of Length 0

15. TN1 transmits Neighbor Solicitation A with an Option Length set to 0.
16. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*
  Step 2: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.
- *Part B*
  Step 4: The NUT must not transmit any packets corresponding to Neighbor Solicitation B.
- *Part C*
  Step 6: The NUT must not transmit any packets corresponding to Neighbor Solicitation C.
- *Part D*
  Step 8: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.
- *Part E*
  Step 10: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.
- *Part F*
  Step 12: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.
- *Part G*
  Step 14: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.
- *Part H*
  Step 16: The NUT must not transmit any packets corresponding to Neighbor Solicitation A.

Possible Problems:

- None.

Test v6LC.2.1.8: Neighbor Solicitation Processing, No NCE

Purpose: Verify that a node properly updates its neighbor cache upon receipt of neighbor solicitations when there is no NCE exists for that neighbor.

References:

- [ND] – Sections 7.2.3 and 7.2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A valid Neighbor Solicitation that does not meet any of the following requirements must be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],
- The Target Address is a unicast address for which the node is offering proxy service, or
- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

If the Target Address is tentative, the Neighbor Solicitation should be processed as described in [ADDRCONF]. Otherwise, the following description applies. If the Source Address is not the unspecified address and, on link layers that have addresses, the solicitation includes a Source Link-Layer Address option, then the recipient SHOULD create or update the Neighbor Cache entry for the IP Source Address of the solicitation. If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE as specified in Section 7.3.3. If an entry exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry's reachability state MUST be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE’s are in state No NCE. The common cleanup procedure is performed after each part.

Neighbor Solicitation A	Neighbor Solicitation B
IPv6 Header Next Header: 58 Destination Address: NUT’s Link-local Address Source Address: TN1’s Link-local Address	IPv6 Header Next Header: 58 Destination Address: NUT’s Solicited-node Multicast Link-local Address Source Address: TN1’s Link-local Address

Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address	Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address

Procedure:

Part A: Unicast Neighbor Solicitation

1. TN1 transmits Neighbor Solicitation A.
2. TN1 transmits an Echo Request to the NUT.
3. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Part B: Multicast Neighbor Solicitation

4. TN1 transmits Neighbor Solicitation B.
5. TN1 transmits an Echo Request to the NUT.
6. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 3: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **STALE**. The NUT should reply to Neighbor Solicitation A by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.
- *Part B*

  Step 4: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **STALE**. The NUT should reply to Neighbor Solicitation B by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.

Possible Problems:

- None.

Test v6LC.2.1.9: Neighbor Solicitation Processing, NCE State INCOMPLETE

Purpose: Verify that a node properly updates its neighbor cache upon receipt of neighbor solicitations when the NCE of the neighbor is in state INCOMPLETE.

References:

- [ND] – Sections 7.2.3 and 7.2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A valid Neighbor Solicitation that does not meet any of the following requirements must be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],
- The Target Address is a unicast address for which the node is offering proxy service, or
- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

If the Target Address is tentative, the Neighbor Solicitation should be processed as described in [ADDRCONF]. Otherwise, the following description applies. If the Source Address is not the unspecified address and, on link layers that have addresses, the solicitation includes a Source Link-Layer Address option, then the recipient SHOULD create or update the Neighbor Cache entry for the IP Source Address of the solicitation. If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE as specified in Section 7.3.3. If an entry exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry's reachability state MUST be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE’s are in state No NCE. The common cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 58 Source Address: TN1’s Link-local Address Destination Address: NUT’s Link-local Address
ICMPv6 Echo Request

Neighbor Solicitation B	Neighbor Solicitation C
IPv6 Header Next Header: 58 Destination Address: NUT's Link-local Address Source Address: TN1's Link-local Address	IPv6 Header Next Header: 58 Destination Address: NUT's Solicited-node Multicast Link-local Address Source Address: TN1's Link-local Address
Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address	Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address

Procedure:

Part A: Unicast Neighbor Solicitation

1. TN1 transmits Echo Request A.
2. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.
3. TN1 transmits Neighbor Solicitation B.
4. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.
5. TN1 transmits an Echo Request to the NUT.
6. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Part B: Multicast Neighbor Solicitation

7. TN1 transmits Echo Request A.
8. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.
9. TN1 transmits Neighbor Solicitation C.
10. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.
11. TN1 transmits an Echo Request to the NUT.
12. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 4: After receiving TN1's Neighbor Solicitation, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **STALE** and update its link-layer address for TN1 accordingly. The NUT should reply to Neighbor Solicitation B by sending a Neighbor Advertisement.
  Step 6: The NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a Unicast Neighbor Solicitation to TN1.
- *Part B*

  Step 8: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 10: After receiving TN1's Neighbor Solicitation, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **STALE** and update its

link-layer address for TN1 accordingly. The NUT should reply to Neighbor Solicitation C by sending a Neighbor Advertisement.
Step 12: The NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a Unicast Neighbor Solicitation to TN1.

Possible Problems:

- None.

Test v6LC.2.1.10: Neighbor Solicitation Processing, NCE State REACHABLE

Purpose: Verify that a node properly updates its neighbor cache upon receipt of neighbor solicitations when the NCE of the neighbor is in state REACHABLE.

References:

- [ND] – Sections 7.2.3 and 7.2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A valid Neighbor Solicitation that does not meet any of the following requirements must be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],
- The Target Address is a unicast address for which the node is offering proxy service, or
- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

If the Target Address is tentative, the Neighbor Solicitation should be processed as described in [ADDRCONF]. Otherwise, the following description applies. If the Source Address is not the unspecified address and, on link layers that have addresses, the solicitation includes a Source Link-Layer Address option, then the recipient SHOULD create or update the Neighbor Cache entry for the IP Source Address of the solicitation. If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE as specified in Section 7.3.3. If an entry exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry's reachability state MUST be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Advertisement B

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1 Target Address: TN1's Link-local Address

Neighbor Solicitation C	Neighbor Solicitation D
IPv6 Header Next Header: 58 Destination Address: NUT's Link-local Address Source Address: TN1's Link-local Address	Ipv6 Header Next Header: 58 Destination Address: NUT's Solicited-node Multicast Link-local Address Source Address: TN1's Link-local Address
Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address	Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address

Procedure:

Part A: Unicast Neighbor Solicitation with the same SLLA

1. TN1 transmits Echo Request A.
2. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
3. TN1 transmits a solicited Neighbor Advertisement B.
4. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
5. TN1 transmits Echo Request A.
6. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
7. TN1 transmits Neighbor Solicitation C.
8. TN1 transmits an Echo Request to the NUT.
9. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Part B: Unicast Neighbor Solicitation with a different SLLA

10. TN1 transmits Echo Request A.
11. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
12. TN1 transmits a solicited Neighbor Advertisement B.

13. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
14. TN1 transmits Echo Request A.
15. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
16. TN1 transmits Neighbor Solicitation C with a different address as the Source Link-layer Address.
17. TN1 transmits an Echo Request to the NUT.
18. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Part C: Multicast Neighbor Solicitation with the same SLLA

19. TN1 transmits Echo Request A.
20. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
21. TN1 transmits a solicited Neighbor Advertisement B.
22. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
23. TN1 transmits Echo Request A.
24. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
25. TN1 transmits Neighbor Solicitation D.
26. TN1 transmits an Echo Request to the NUT.
27. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Part D: Multicast Neighbor Solicitation with a different SLLA

28. TN1 transmit Echo Request A.
29. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
30. TN1 transmits a solicited Neighbor Advertisement B.
31. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
32. TN1 transmits Echo Request A.
33. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
34. TN1 transmits Neighbor Solicitation D with a different address as the Source Link-layer Address.
35. TN1 transmits an Echo Request to the NUT.
36. Check the NCE of TN1 on the NUT and observe packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 4: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 6: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 9: The NUT should not update the NCE of TN1, the NUT should reply to Neighbor Solicitation C by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and should stay in state **REACHABLE**.

- *Part B*

  Step 11: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 13: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.

Step 15: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
Step 18: The NUT should update the NCE of TN1 to state **STALE** and update TN1's Link-layer address to its new Link-layer address from the received Neighbor Solicitation C. The NUT should reply to Neighbor Solicitation C by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 with the Target set the new Link-Layer address of TN1.

- *Part C*

  Step 20: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 22: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 24: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 27: The NUT should not update the NCE of TN1, the NUT should reply to Neighbor Solicitation D by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and should stay in state **REACHABLE**.

- *Part D*

  Step 29: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 31: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 33: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 36: The NUT should update the NCE of TN1 to state **STALE** and update TN1's Link-layer address to its new Link-layer address from the received Neighbor Solicitation C. The NUT should reply to Neighbor Solicitation C by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 with the Target set to the new Link-Layer address of TN1.

Possible Problems:

- None.

Test v6LC.2.1.11: Neighbor Solicitation Processing, NCE State STALE

Purpose: Verify that a node properly updates its neighbor cache upon receipt of neighbor solicitations when the NCE of the neighbor is in state STALE.

References:

- [ND] – Sections 7.2.3 and 7.2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A valid Neighbor Solicitation that does not meet any of the following requirements must be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],
- The Target Address is a unicast address for which the node is offering proxy service, or
- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

If the Target Address is tentative, the Neighbor Solicitation should be processed as described in [ADDRCONF]. Otherwise, the following description applies. If the Source Address is not the unspecified address and, on link layers that have addresses, the solicitation includes a Source Link-Layer Address option, then the recipient SHOULD create or update the Neighbor Cache entry for the IP Source Address of the solicitation. If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE as specified in Section 7.3.3. If an entry exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry's reachability state MUST be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Advertisement B

IPv6 Header Next Header: 58 Source Address: TN1’s Link-local Address Destination Address: NUT’s Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1 Target Address: TN1’s Link-local Address

Neighbor Solicitation C	Neighbor Solicitation D
IPv6 Header Next Header: 58 Destination Address: NUT’s Link-local Address Source Address: TN1’s Link-local Address	Ipv6 Header Next Header: 58 Destination Address: NUT’s Solicited-node Multicast Link-local Address Source Address: TN1’s Link-local Address
Neighbor Solicitation Target Address: NUT’s Link-local Address Source Link-Layer Address: TN1’s Ethernet address	Neighbor Solicitation Target Address: NUT’s Link-local Address Source Link-Layer Address: TN1’s Ethernet address

Procedure:

Part A: Unicast Neighbor Solicitation with the same SLLA

1. TN1 transmits Echo Request A.
2. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
3. TN1 transmits a solicited Neighbor Advertisement B.
4. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
5. TN1 transmits Echo Request A.
6. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
7. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds.
8. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
9. TN1 transmits Neighbor Solicitation C.
10. TN1 transmits an Echo Request to the NUT.
11. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part B: Unicast Neighbor Solicitation with a different SLLA

12. TN1 transmits Echo Request A.

13. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
14. TN1 transmits a solicited Neighbor Advertisement B.
15. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
16. TN1 transmits Echo Request A.
17. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
18. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds.
19. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
20. TN1 transmits Neighbor Solicitation C with a different address as the Source Link-layer Address.
21. TN1 transmits an Echo Request to the NUT.
22. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part C: Multicast Neighbor Solicitation with the same SLLA

23. TN1 transmits Echo Request A.
24. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
25. TN1 transmits a solicited Neighbor Advertisement B.
26. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
27. TN1 transmits Echo Request A.
28. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
29. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds.
30. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
31. TN1 transmits Neighbor Solicitation D.
32. TN1 transmits an Echo Request to the NUT.
33. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part D: Multicast Neighbor Solicitation with a different SLLA

34. TN1 transmit Echo Request A.
35. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
36. TN1 transmits a solicited Neighbor Advertisement B.
37. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
38. TN1 transmits Echo Request A.
39. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
40. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds;
41. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
42. TN1 transmits Neighbor Solicitation D with a different address as the Source Link-layer Address.
43. TN1 transmits an Echo Request to the NUT.
44. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 4: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 6: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 8: The NUT should update the NCE of TN1 to state **STALE**. (See Note in Section 2 title page.)

Step 11: The NUT should not update the NCE of TN1 and should stay in state **STALE**. The NUT should reply to the Neighbor Solicitation by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.

- *Part B*

  Step 13: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 15: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 17: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 19: The NUT should update the NCE of TN1 to state **STALE**. (See Note in Section 2 title page.)
  Step 22: The NUT should update TN1's Link-layer address to its new link-layer address from the received Neighbor Solicitation C. The NUT should not update the NCE of TN1 and should stay in state **STALE**. The NUT should reply to the Neighbor Solicitation by sending a Neighbor Advertisement to TN1's new Link-Layer address. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 using the new link-layer address as the Target.

- *Part C*

  Step 24: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 26: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE** and update its link-layer address for TN1 accordingly.
  Step 28: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 30: The NUT should update the NCE of TN1 to state **STALE**. (See Note in Section 2 title page.)
  Step 33: The NUT should not update the NCE of TN1 and should stay in state **STALE**. The NUT should not update the NCE of TN1. The NUT should reply to the Neighbor Solicitation by sending a Neighbor Advertisement. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.

- *Part D*

  Step 35: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 37: After receiving TN1's Neighbor Advertisement, the NUT should send its queued Echo Reply to TN1. The NUT should then update the NCE of TN1 to state **REACHABLE**

and update its link-layer address for TN1 accordingly.
Step 39: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
Step 41: The NUT should update the NCE of TN1 to state **STALE**. (See Note in Section 2 title page.)
Step 44: The NUT should update TN1's Link-layer address to its new link-layer address from the received Neighbor Solicitation D. The NUT should not update the NCE of TN1 and should stay in state **STALE**. The NUT should reply to the Neighbor Solicitation by sending a Neighbor Advertisement to TN1's new Link-Layer address. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 using the new link-layer address as the Target.

Possible Problems:

- This test will be inaccurate if the NUT Failed test V6LC.2.1.7 testing (REACHABLE_TIME*MAX_RANDOM_FACTOR).

Test v6LC.2.1.12: Neighbor Solicitation Processing, NCE State PROBE

Purpose: Verify that a node properly updates its neighbor cache upon receipt of neighbor solicitations when the NCE of the neighbor is in state Probe.

References:

- [ND] – Sections 7.2.3 and 7.2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A valid Neighbor Solicitation that does not meet any of the following requirements must be silently discarded:

- The Target Address is a "valid" unicast or anycast address assigned to the receiving interface [ADDRCONF],
- The Target Address is a unicast address for which the node is offering proxy service, or
- The Target Address is a "tentative" address on which Duplicate Address Detection is being performed [ADDRCONF].

If the Target Address is tentative, the Neighbor Solicitation should be processed as described in [ADDRCONF]. Otherwise, the following description applies. If the Source Address is not the unspecified address and, on link layers that have addresses, the solicitation includes a Source Link-Layer Address option, then the recipient SHOULD create or update the Neighbor Cache entry for the IP Source Address of the solicitation. If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE as specified in Section 7.3.3. If an entry exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry's reachability state MUST be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Advertisement B

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 0 Override flag: 1 Target Address: TN1's Link-local Address

Neighbor Solicitation C	Neighbor Solicitation D
IPv6 Header Next Header: 58 Destination Address: NUT's Link-local Address Source Address: TN1's Link-local Address	Ipv6 Header Next Header: 58 Destination Address: NUT's Solicited-node Multicast Link-local Address Source Address: TN1's Link-local Address
Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address	Neighbor Solicitation Target Address: NUT's Link-local Address Source Link-Layer Address: TN1's Ethernet address

Procedure:

Part A: Unicast Neighbor Solicitation with the same SLLA

1. TN1 transmits Packet A to the NUT.
2. TN1 transmits Neighbor Advertisement B to the NUT after receiving any Neighbor Solicitations from the NUT.
3. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
4. Wait (DELAY_FIRST_PROBE_TIME) seconds.
5. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
6. TN1 transmits Neighbor Solicitation C.
7. TN1 transmits an Echo Request to the NUT.
8. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part B: Unicast Neighbor Solicitation with a different SLLA

9. TN1 transmits Packet A to the NUT.
10. TN1 transmits Neighbor Advertisement B to the NUT after receiving any Neighbor Solicitations from the NUT.
11. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
12. Wait (DELAY_FIRST_PROBE_TIME) seconds.
13. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
14. TN1 transmits Neighbor Solicitation C with a different address as the Source Link-layer Address.
15. TN1 transmits an Echo Request to the NUT.
16. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part C: Multicast Neighbor Solicitation with the same SLLA

17. TN1 transmits Packet A to the NUT.
18. TN1 transmits Neighbor Advertisement B to the NUT after receiving any Neighbor Solicitations from the NUT
19. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
20. Wait (DELAY_FIRST_PROBE_TIME) seconds.
21. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
22. TN1 transmits Neighbor Solicitation D.
23. TN1 transmits an Echo Request to the NUT.
24. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Part D: Multicast Neighbor Solicitation with a different SLLA

25. TN1 transmits Packet A to the NUT.
26. TN1 transmits Neighbor Advertisement B to the NUT after receiving any Neighbor Solicitations from the NUT
27. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
28. Wait (DELAY_FIRST_PROBE_TIME) seconds.
29. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT..
30. TN1 transmits Neighbor Solicitation D with a different address as the Source Link-layer Address.
31. TN1 transmits an Echo Request to the NUT.
32. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 3: The NUT should update the NCE of TN1 to state **STALE**. After receiving the Echo Request from TN1, the NUT should send a Reply and enter state **DELAY.**
  Step 5: After DELAY_FIRST_PROBE_TIME, the NUT should transition to state **PROBE** by sending a unicast Neighbor Solicitation to TN1.
  Step 8: The NUT should not update the state of TN1's NCE after sending its queued Neighbor Advertisement and Echo Reply and should stay in state **PROBE**. The NUT should retransmit its unicast Neighbor Solicitation to TN1.
- *Part B*

  Step 11: The NUT should update the NCE of TN1 to state **STALE**. After receiving the Echo Request from TN1, the NUT should send an Echo Reply and enter state **DELAY.**
  Step 13: After DELAY_FIRST_PROBE_TIME, the NUT should transition to state **PROBE** by sending a unicast Neighbor Solicitation to TN1.
  Step 16: The NUT should update TN1's Link-layer address to its new link-layer address from the received Neighbor Solicitation C and MUST update the state of TN1's NCE to **STALE**. The NUT should reply to the Neighbor Solicitation by sending a Neighbor

Advertisement using TN1's new Link-Layer address. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 using the new Link-layer address as the Target.

- *Part C*

  Step 19: The NUT should update the NCE of TN1 to state **STALE**. After receiving the Echo Request from TN1, the NUT should send a Reply and enter state **DELAY.**
  Step 21: After DELAY_FIRST_PROBE_TIME, the NUT should transition to state **PROBE** by sending a unicast Neighbor Solicitation to TN1.
  Step 24: The NUT should not update the state of TN1's NCE after sending it's queued Neighbor Advertisement and Echo Reply and should stay in state **PROBE**. The NUT should retransmit its unicast Neighbor Solicitation to TN1.
- *Part D*

  Step 27: The NUT should update the NCE of TN1 to state **STALE**. After receiving the Echo Request from TN1, the NUT should send a Reply and enter state **DELAY.**
  Step 29: After DELAY_FIRST_PROBE_TIME, the NUT should transition to state **PROBE** by sending a unicast Neighbor Solicitation to TN1.
  Step 32: The NUT should update TN1's Link-layer address to its new link-layer address from the received Neighbor Solicitation C and MUST update the state of TN1's NCE to **STALE**. The NUT should reply to the Neighbor Solicitation by sending a Neighbor Advertisement using TN1's new Link-Layer address. After responding to the Neighbor Solicitation, the NUT should respond to the Echo Request by sending an Echo Reply and set the state of the TN1's Entry to **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1 using the new Link-layer address as the Target.

Possible Problems:

- None.

Test v6LC.2.1.13: Neighbor Solicitation Processing, Anycast (Routers Only)

Purpose: Verify that a router properly processes a Neighbor Solicitation for an anycast address.

References:

- [ND] – Sections 7.2.3 and 7.2.4
- [ADDR] – Section 2.6.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Nodes that have an anycast address assigned to an interface treat them exactly the same as if they were unicast addresses with two exceptions. First, Neighbor Advertisements sent in response to a Neighbor Solicitation SHOULD be delayed by a random time between 0 and MAX_ANYCAST_DELAY_TIME to reduce the probability of network congestion. Second, the Override flag in Neighbor Advertisements SHOULD be set to 0, so that when multiple advertisements are received, the first received advertisement is used rather than the most recently received advertisement.

The Subnet-Router anycast address is predefined. Its format is as follows:

```
|                         n bits                 |   128-n bits   |
+------------------------------------------------+----------------+
|                   subnet prefix                | 00000000000000 |
+------------------------------------------------+----------------+
```

The "subnet prefix" in an anycast address is the prefix which identifies a specific link. This anycast address is syntactically the same as a unicast address for an interface on the link with the interface identifier set to zero.

Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet. All routers are required to support the Subnet-Router anycast addresses for the subnets which they have interfaces.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

1. Configure the RUT to advertise prefix X on Link B.
2. Configure an address with prefix X on the RUT.

Procedure:

1. TN1 transmits a Neighbor Solicitation to the RUT's Subnet-Router anycast address.
2. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT should respond to TN1 by sending a Neighbor Advertisement between 0 and MAX_ANYCAST_DELAY_TIME after it receives the Neighbor Solicitation. The RUT's Neighbor Advertisement should contain a value of 0 in the override flag field.

Possible Problems:

- None.

Test v6LC.2.1.14: Invalid Neighbor Advertisement Handling

Purpose: Verify that a node takes the proper actions upon receipt of an invalid Neighbor Advertisement.

References:

- [ND] – Section 7.1.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node MUST silently discard any received Neighbor Advertisement messages that do not satisfy all of the following validity checks:

- IP Hop Limit has a value of 255
- If Message includes an Authentication Header, it is authenticated correctly
- ICMP Checksum is valid
- ICMP code = 0
- ICMP length is 24 or more octets
- Target Address is not a multicast address
- If the IP Destination Address is multicast, the Solicited Flag = 0
- All included options have a length > 0

The contents of any defined options that are not specified to be use with Neighbor Advertisement messages MUST be ignored and the packet processed as normal. The only defined option that may appear is the Target Link-Layer Address option.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Neighbor Advertisement A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: all-nodes multicast address
Neighbor Advertisement ICMP Code: 0 ICMP Checksum: Valid Router flag: 0 Solicited flag: 0 Override flag: 1 Target Address: TN1's link-

local address TLLOPT: TN1’s MAC address

Procedure:

Part A: NUT receives invalid NA (Solicited Flag ==1)

1. TN1 transmits an Echo Request to the NUT.
2. Observe the packets transmitted by the NUT.
3. TN1 transmits Neighbor Advertisement A with the Solicited flag set to 1.
4. Observe packet captures on Link B.

Part B: NUT receives invalid NA (Hop Limit == 254)

5. TN1 transmits an Echo Request to the NUT.
6. Observe the packets transmitted by the NUT.
7. Configure TN1 to transmit Neighbor Advertisement A with the Hop Limit set to 254.
8. Observe packet captures on Link B.

Part C: NUT receives invalid NA (Invalid Checksum)

9. TN1 transmits an Echo Request to the NUT.
10. Observe the packets transmitted by the NUT.
11. Configure TN1 to transmit Neighbor Advertisement A with an invalid checksum.
12. Observe packet captures on Link B.

Part D: NUT receives invalid NA (ICMP code!= zero)

13. TN1 transmits an Echo Request to the NUT.
14. Observe the packets transmitted by the NUT.
15. Configure TN1 to transmit Neighbor Advertisement A with the ICMP code set to 1.
16. Observe packet captures on Link B.

Part E: NUT receives invalid NA (ICMP length < 24 octets)

17. TN1 transmits an Echo Request to the NUT.
18. Observe the packets transmitted by the NUT.
19. Configure TN1 to transmit Neighbor Advertisement A with the ICMP length set to 16.
20. Observe packet captures on Link B.

Part F: NUT receives invalid NA (target == multicast address)

21. TN1 transmits an Echo Request to the NUT.
22. Observe the packets transmitted by the NUT.
23. Configure TN1 to transmit Neighbor Advertisement A with the Target Address set to the solicited multicast of TN1’s link-local address.
24. Observe packet captures on Link B.

Part G: NUT receives invalid NA (option length ==zero)

25. TN1 transmits an Echo Request to the NUT.
26. Observe the packets transmitted by the NUT.
27. Configure TN1 to transmit Neighbor Advertisement A with the Option length set to 0.
28. Observe packet captures on Link B.

Observable Results:

- *Part A*

  Step 2: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 4: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.
- *Part B*

  Step 6: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 8: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.
- *Part C*

  Step 10: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 12: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.
- *Part D*

  Step 14: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 16: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.
- *Part E*

  Step 18: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 20: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.
- *Part F*

  Step 22: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 24: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.
- *Part G*

  Step 26: The NUT should transmit a Neighbor Solicitation to TN1's solicited-node multicast address.

  Step 28: The NUT should ignore the Neighbor Advertisement sent by TN1 and should continue to transmit Neighbor Solicitations to TN1's solicited-node multicast address.

Possible Problems:

- None.

Test v6LC.2.1.15: Neighbor Advertisement Processing, No NCE

Purpose: Verify that a node silently discards a Neighbor Advertisement if the target does not have a Neighbor Cache entry.

References:

- [ND] – Section 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a valid Neighbor Advertisement is received (either solicited or unsolicited), the Neighbor Cache is searched for the target's entry. If no entry exists, the advertisement SHOULD be silently discarded. There is no need to create an entry if none exists, since the recipient has apparently not initiated any communication with the target.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Neighbor Advertisement A

IPv6 Header Next Header: 58
Neighbor Advertisement Solicited flag: 0 Override flag: 0 Target Link-layer Option

Neighbor Advertisement B

IPv6 Header Next Header: 58
Neighbor Advertisement Solicited flag: 0 Override flag: 1 Target Link-layer Option

Neighbor Advertisement C

IPv6 Header Next Header: 58
Neighbor Advertisement Solicited flag: 1 Override flag: 0 Target Link-layer Option

Neighbor Advertisement D

IPv6 Header Next Header: 58
Neighbor Advertisement Solicited flag: 1 Override flag: 1 Target Link-layer Option

Procedure:

Part A: Receiving NA with S = 0, O = 0, and TLLA

1. TR1 transmits Neighbor Advertisement A.
2. Observe the packets transmitted by the NUT and the NC on the NUT.
3. TN1 transmits an Echo Request to the NUT.
4. Observe the packets transmitted by the NUT and the NC on the NUT.

Part B: Receiving NA with S = 0, O = 1, and TLLA

5. TR1 transmits Neighbor Advertisement B.
6. Observe the packets transmitted by the NUT and the NC on the NUT.
7. TN1 transmits an Echo Request to the NUT.
8. Observe the packets transmitted by the NUT and the NC on the NUT.

Part C: Receiving NA with S = 1, O = 0, and TLLA

9. TR1 transmits Neighbor Advertisement C.
10. Observe the packets transmitted by the NUT and the NC on the NUT.
11. TN1 transmits an Echo Request to the NUT.
12. Observe the packets transmitted by the NUT and the NC on the NUT.

Part D: Receiving NA with S = 1, O = 1, and TLLA

13. TR1 transmits Neighbor Advertisement D.
14. Observe the packets transmitted by the NUT and the NC on the NUT.
15. TN1 transmits an Echo Request to the NUT.
16. Observe the packets transmitted by the NUT and the NC on the NUT.

Part E: Receiving NA with S = 0, O = 0, and NO TLLA

17. TR1 transmits Neighbor Advertisement A without the Target Link-layer Address Option.
18. Observe the packets transmitted by the NUT and the NC on the NUT.
19. TN1 transmits an Echo Request to the NUT.
20. Observe the packets transmitted by the NUT and the NC on the NUT.

Part F: Receiving NA with S = 0, O = 1, and NO TLLA

21. TR1 transmits Neighbor Advertisement B without the Target Link-layer Address Option.
22. Observe the packets transmitted by the NUT and the NC on the NUT.
23. TN1 transmits an Echo Request to the NUT.

24. Observe the packets transmitted by the NUT and the NC on the NUT.

Part G: Receiving NA with S = 1, O = 0, and NO TLLA

25. TR1 transmits Neighbor Advertisement C without the Target Link-layer Address Option.
26. Observe the packets transmitted by the NUT and the NC on the NUT.
27. TN1 transmits an Echo Request to the NUT.
28. Observe the packets transmitted by the NUT and the NC on the NUT.

Part H: Receiving NA with S = 1, O = 1, and NO TLLA

29. TR1 transmits Neighbor Advertisement D without the Target Link-layer Address Option.
30. Observe the packets transmitted by the NUT and the NC on the NUT.
31. TN1 transmits an Echo Request to the NUT.
32. Observe the packets transmitted by the NUT and the NC on the NUT.

Observable Results:

- *Parts A-H*

  For each part, after receiving the Neighbor Advertisement from TN1, the NUT should not transmit any packets and no NCE's should be created for TR1. After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.

Possible Problems:

- None.

Test v6LC.2.1.16: Neighbor Advertisement Processing, NCE State INCOMPLETE

Purpose: Verify that a node properly updates its Neighbor Cache from the INCOMPLETE state upon receipt of a Neighbor Advertisement.

References:

- [ND] – Section 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a Neighbor Cache entry is in the INCOMPLETE state the receipt of a Neighbor Advertisement causes the state of the entry to change as follows, based upon the Solicited and Override flags of the advertisement:

Solicited flag	Override flag	New State	Update Link-Layer Address
set	set	REACHABLE	yes
set	clear	REACHABLE	yes
clear	set	STALE	yes
clear	clear	STALE	yes

If the target's Neighbor Cache entry is in the INCOMPLETE state when the advertisement is received, one of two things happens. If the link layer has addresses and no Target Link-Layer address option is included, the receiving node SHOULD silently discard the received advertisement.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Packet A
IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Adv. B	Neighbor Adv. C	Neighbor Adv. D	Neighbor Adv. E
IPv6 Header Next Header: 58	IPv6 Header Next Header: 58	IPv6 Header Next Header: 58	IPv6 Header Next Header: 58
Neighbor Adv. Solicited flag: 1 Override flag: 1	Neighbor Adv. Solicited flag: 1 Override flag: 0	Neighbor Adv. Solicited flag: 0 Override flag: 1	Neighbor Adv. Solicited flag: 0 Override flag: 0

Procedure:

Part A: Receiving NA with S = 1 and O = 1

1. TN1 transmits Packet A.
2. Observe the packets transmitted by the NUT.
3. TN1 transmits Neighbor Advertisement B.
4. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.
5. TN1 transmits an Echo Request.
6. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.

Part B: Receiving NA with S = 1 and O = 0

7. TN1 transmits Packet A.
8. Observe the packets transmitted by the NUT.
9. TN1 transmits Neighbor Advertisement C.
10. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.
11. TN1 transmits an Echo Request.
12. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.

Part C: Receiving NA with S = 0 and O = 1

13. TN1 transmits Packet A.
14. Observe the packets transmitted by the NUT.
15. TN1 transmits Neighbor Advertisement D.
16. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.
17. TN1 transmits an Echo Request.
18. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.

Part D: Receiving NA with S = 0 and O = 0

19. TN1 transmits Packet A.
20. Observe the packets transmitted by the NUT.
21. TN1 transmits Neighbor Advertisement E.
22. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.
23. TN1 transmits an Echo Request.
24. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.

Part E: Receiving NA without Target Link-Layer Address Option

25. TN1 transmits Packet A.
26. Observe the packets transmitted by the NUT.
27. TN1 transmits a Neighbor Advertisement without any Target Link-Layer Address Option.
28. Observe the packets transmitted by the NUT and the NCE of TN1 on the NUT.

Observable Results:

- *Part A*

Step 2: After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
Step 4: After receiving the Neighbor Advertisement from TN1, the NUT should send the queued Echo Reply to TN1 and update its NCE of TN1 with the received Target Link-layer Address and change the state of the NCE to **REACHABLE**.
Step 6: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.

- *Part B*

  Step 8: After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 10: After receiving the Neighbor Advertisement from TN1, the NUT should send the queued Echo Reply to TN1 and update its NCE of TN1 with the received Target Link-layer Address and change the state of the NCE to **REACHABLE**.
  Step 12: Because the NUT is in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.

- *Part C*

  Step 14: After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 16: After receiving the Neighbor Advertisement from TN1, the NUT should send the queued Echo Reply to TN1 and update its NCE of TN1 with the received Target Link-layer Address and change the state of the NCE to **STALE**.
  Step 18: Because the NUT is in state **STALE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.

- *Part D*

  Step 20: After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 22: After receiving the Neighbor Advertisement from TN1, the NUT should send the queued Echo Reply to TN1 and update its NCE of TN1 with the received Target Link-layer Address and change the state of the NCE to **STALE**.
  Step 24: Because the NUT is in state **STALE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.

- *Part E*

  Step 26: After receiving the Echo Request from TN1, the NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 28: The NUT should ignore the NA transmitted by TN1. There should be no change in the neighbor cache for TN1 as it should stay in state **INCOMPLETE**. The NUT should continue to send multicast Neighbor Solicitation to TN1.

Possible Problems:

- None.

Test v6LC.2.1.17: Neighbor Advertisement Processing, NCE State REACHABLE

Purpose: Verify that a node properly updates its Neighbor Cache from the REACHABLE state upon receipt of a Neighbor Advertisement.

References:

- [ND] – Section 7.3.3 and 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a Neighbor Cache entry is in the REACHABLE state, the receipt of a Neighbor Advertisement causes the state of the entry to change as follows, based upon the Solicited and Override flags of the advertisement and whether the new link-layer address is equal to the previously cached value:

Destination	Solicited flag	Override flag	TLLA	New State	Update Link-Layer Address	Part
Unicast	clear	clear	none	REACHABLE	no	A
Unicast	clear	set	none	REACHABLE	no	B
Unicast	set	clear	none	REACHABLE	no	C
Unicast	set	set	none	REACHABLE	no	D
Unicast	clear	clear	same	REACHABLE	no	E
Unicast	clear	set	same	REACHABLE	no	F
Unicast	set	clear	same	REACHABLE	no	G
Unicast	set	set	same	REACHABLE	no	H
Unicast	clear	clear	different	STALE	no	I
Unicast	clear	set	different	STALE	yes	J
Unicast	set	clear	different	STALE	no	K
Unicast	set	set	different	REACHABLE	yes	L
Multicast	clear	clear	same	REACHABLE	no	M
Multicast	clear	set	same	REACHABLE	no	N
Multicast	clear	clear	different	STALE	no	O
Multicast	clear	set	different	STALE	yes	P

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Echo Request A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Adv. (A-H)

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: see table
Neighbor Adv. Solicited flag: see table Override flag: see table
Target LLA Option: see table

Procedure:

1. TN1 transmits Echo Request A.
2. Observe the packets transmitted by the NUT and the NCE of TN1.
3. TN1 transmits a solicited Neighbor Advertisement to the NUT.
4. Observe the packets transmitted by the NUT and the NCE of TN1.
5. TN1 transmits Neighbor Advertisement A. The Solicited and Override flags are set according to Part A entry of the table in the discussion above. Similarly, the address in the Target Link Layer Address Option is provided as it is indicated.
6. TN1 transmits an Echo Request.
7. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
8. Perform the common cleanup procedure.
9. Repeat Steps 1 through 7 fifteen times for Parts B through P.

Observable Results:

- *Parts A through P*

  Step 2: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the

Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
Step 4: Because the NUT is now in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
Step 7: The NUT MUST update the state of TN1's NCE and the LLA according to the table in the discussion above. After receiving the Echo Request from TN1 in step 6, the NUT should react according to the following:

Parts A-H and L-N to REACHABLE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
Part L
The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1.

Parts I-K and O-P to STALE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1.
Parts J and P
The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1. The Neighbor Solicitation should use the new link-layer address in the Target field.

Possible Problems:

- None.

Test v6LC.2.1.18: Neighbor Advertisement Processing, NCE State STALE

Purpose: Verify that a node properly updates its Neighbor Cache from the STALE state upon receipt of a Neighbor Advertisement.

References:

- [ND] – Section 7.3.3 and 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a Neighbor Cache entry is in the STALE state, the receipt of a Neighbor Advertisement causes the state of the entry to change as follows, based upon the Solicited and Override flags of the advertisement, whether the new link-layer address is provided, and whether it is equal to the previously cached value:

Destination	**Solicited flag**	**Override flag**	**TLLA**	**New State**	**Update Link-Layer Address**	**Part**
Unicast	clear	clear	none	STALE	no	A
Unicast	clear	set	none	STALE	no	B
Unicast	set	clear	none	REACHABLE	no	C
Unicast	set	set	none	REACHABLE	no	D
Unicast	clear	clear	same	STALE	no	E
Unicast	clear	set	same	STALE	no	F
Unicast	set	clear	same	REACHABLE	no	G
Unicast	set	set	same	REACHABLE	no	H
Unicast	clear	clear	different	STALE	no	I
Unicast	clear	set	different	STALE	yes	J
Unicast	set	clear	different	STALE	no	K
Unicast	set	set	different	REACHABLE	yes	L
Multicast	clear	clear	same	STALE	no	M
Multicast	clear	set	same	STALE	no	N
Multicast	clear	clear	different	STALE	no	O
Multicast	clear	set	different	STALE	yes	P

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Echo Request A IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Adv. (A-P)

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: see table
Neighbor Adv. Solicited flag: see table Override flag: see table
Target LLA Option: see table

Procedure:

1. TN1 transmits Echo Request A.
2. Observe the packets transmitted by the NUT and the NCE of TN1.
3. TN1 transmits a solicited Neighbor Advertisement to the NUT.
4. Observe the packets transmitted by the NUT and the NCE of TN1.
5. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds.
6. Check the NCE of TN1 on the NUT.
7. TN1 transmits Neighbor Advertisement A. The Solicited and Override flags are set according to Part A entry of the table in the discussion above. Similarly, the address in the Target Link Layer Address Option is provided as it is indicated.
8. TN1 transmits an Echo Request.
9. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
10. Perform the common cleanup procedure.
11. Repeat Steps 1 through 9 fifteen times for Parts B through P.

Observable Results:

- *Parts A through P*

  Step 2: The NUT should create a Neighbor Cache Entry for TN1 and set the state of the Entry to **INCOMPLETE**. The NUT should send a multicast Neighbor Solicitation to TN1.
  Step 4: Because the NUT is now in state **REACHABLE**, after receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1.
  Step 6: The NUT should change the state of TN1's NCE to **STALE**. (See Note in Section 2 title page.)
  Step 9: The NUT MUST update the state of TN1's NCE and the LLA according to the table in the discussion above. After receiving the Echo Request from TN1 in step 8, the NUT should react according to the following:

Parts C,D,G,H and L to REACHABLE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1. **Part L** The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1.

Parts A,B,E,F,I-K, and M-P to STALE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1. **Parts J and P** The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1. The Neighbor Solicitation should use the new link-layer address in the Target field.

Possible Problems:

- This test will be inaccurate if the NUT Failed test V6LC.2.1.7 testing (REACHABLE_TIME*MAX_RANDOM_FACTOR).

Test v6LC.2.1.19: Neighbor Advertisement Processing, NCE State PROBE

Purpose: Verify that a node properly updates its Neighbor Cache from the PROBE state upon receipt of a Neighbor Advertisement.

References:

- [ND] – Section 7.3.3 and 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a Neighbor Cache entry is in the PROBE state, the receipt of a Neighbor Advertisement causes the state of the entry to change as follows, based upon the Solicited and Override flags of the advertisement, whether the new link-layer address is provided, and whether it is equal to the previously cached value:

Destination	Solicited flag	Override flag	TLLA	New State	Update Link-LayerAddress	Part
Unicast	clear	clear	none	PROBE	no	A
Unicast	clear	set	none	PROBE	no	B
Unicast	set	clear	none	REACHABLE	no	C
Unicast	set	set	none	REACHABLE	no	D
Unicast	clear	clear	same	PROBE	no	E
Unicast	clear	set	same	PROBE	no	F
Unicast	set	clear	same	REACHABLE	no	G
Unicast	set	set	same	REACHABLE	no	H
Unicast	clear	clear	different	PROBE	no	I
Unicast	clear	set	different	STALE	yes	J
Unicast	set	clear	different	PROBE	no	K
Unicast	set	set	different	REACHABLE	yes	L
Multicast	clear	clear	same	PROBE	no	M
Multicast	clear	set	same	PROBE	no	N
Multicast	clear	clear	different	PROBE	no	O
Multicast	clear	set	different	STALE	yes	P

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Echo Request A

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
ICMPv6 Echo Request

Neighbor Adv. (A-P)

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: see table
Neighbor Adv. Solicited flag: see table Override flag: see table
Target LLA Option: see table

Neighbor Adv. Q

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: NUT's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 0 Override flag: 1 Target Address: TN1's Link-local Address

Procedure:

1. TN1 transmits Echo Request A to the NUT.
2. TN1 transmits Neighbor Advertisement Q to the NUT.
3. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
4. Wait (DELAY_FIRST_PROBE_TIME) seconds.
5. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
6. TN1 transmits Neighbor Advertisement A. The Solicited and Override flags are set according to Part A entry of the table in the discussion above. Similarly, the address in the Target Link Layer Address Option is provided as it is indicated.
7. Skip this step for Parts A, B, E, F, I, K, M, N and O; TN1 transmits an Echo Request.
8. Check the NCE of TN1 on the NUT and observe the packets transmitted by the NUT.
9. Perform the common cleanup procedure.
10. Repeat Steps 1 through 9 fifteen times for Parts B through P.

Observable Results:

- *Parts A through P*

  Step 3: The NUT should change the state of TN1's NCE to **STALE**. After receiving the Echo Request from TN1, the NUT should send a Reply and enter state **DELAY.**
  Step 5: The NUT should change the state of TN1's NCE to **PROBE** by transmitting a unicast Neighbor Solicitation to TN1.
  Step 8: The NUT MUST update the state of TN1's NCE and the LLA according to the table in the discussion above. After receiving the Echo Request from TN1 in step 7, the NUT should react according to the following:

  Parts C, D, G, H and L to REACHABLE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TN1. **Part L** The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1.

  Parts J and P to STALE

After receiving the Echo Request from TN1, the NUT should send an Echo Reply. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TN1. The NUT's Echo Reply should be sent to the new updated link-layer destination address of TN1. The Neighbor Solicitation should use the new link-layer address in the Target field.

  Parts A, B, E, F, I, K, and M-0 to PROBE

The NUT should send a unicast Neighbor Solicitation to TN1.

Possible Problems:

- None.

Test v6LC.2.1.20: Neighbor Advertisement Processing, R-bit Change (Hosts Only)

Purpose: Verify that a host takes appropriate actions when a neighbor who is a router starts transmitting Neighbor Advertisements with the Router flag clear.

References:

- [ND] – Section 7.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The IsRouter flag in the cache entry MUST be set based on the Router flag in the received advertisement. In those cases where the IsRouter flag changes from TRUE to FALSE as a result of this update, the host MUST remove that router from the Default Router List and update the Destination Cache entries for all destinations using that neighbor as a router as specified in Section 7.3.3. This is needed to detect when a node that is configured as a router stops forwarding packets due to being configured as a host.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Advertisement

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address
Router Advertisement Router Lifetime: 20 seconds Reachable Time: 100 seconds Retransmit Interval: 1 second Prefix: TR1's Global Prefix

Packet A

IPv6 Header Next Header: 58 Source Address: TN1's off-link Global Address Destination Address: HUT's Global Address
ICMPv6 Echo Request

Neighbor Advertisement A	Neighbor Advertisement B
IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address	IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1	Neighbor Advertisement Router flag: 0 Solicited flag: 0 Override flag: 0

Neighbor Advertisement C	Neighbor Advertisement D
IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address	IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 0 Override flag: 1	Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 0

Procedure:

1. TR1 transmits the Router Advertisement without a Source Link-layer Address Option.
2. TN1 transmits Packet A.
3. Observe the packets transmitted by the HUT.
4. TR1 responds to Neighbor Solicitations from the HUT with a Neighbor Advertisement with the Router, Solicited, and Override flags set.
5. Observe the packets transmitted by the HUT.
6. TR1 transmits Neighbor Advertisement A.
7. TN1 transmits Packet A.
8. Observe the packets transmitted by the HUT.
9. Perform common cleanup procedure.
10. Repeat Steps 1 through 8 three times with Neighbor Advertisement B, C, and D respectively in Step 6.

Observable Results:

Step 3: The HUT should solicit TR1 by transmitting Neighbor Solicitations with a Target Address of TR1's Link-local Address.
Step 5: The HUT should transmit an Echo Reply to Packet A using the TR1 as the first hop.
Step 8: The HUT MUST not transmit an Echo Reply using TR1 as the first hop in response to Packet A in Step 7 and the HUT MUST not transmit multicast NS's with a target set to TR1's link-local address.

Possible Problems:

- None.

Group 2: Router and Prefix Discovery

Scope

The following tests cover Router and Prefix Discovery in IPv6.

Overview

The tests in this group verify that a host properly performs Router and Prefix Discovery.

Test v6LC.2.2.1: Router Solicitations (Hosts Only)

Purpose: Verify that a host sends valid Router Solicitations at the appropriate time.

References:

- [ND] – Section 6.3.7

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When an interface becomes enabled, a host may be unwilling to wait for the next unsolicited Router Advertisement to locate default routers or learn prefixes. To obtain Router Advertisements quickly, a host SHOULD transmit up to MAX_RTR_SOLICITATIONS Router Solicitation messages each separated by at least RTR_SOLICITATION_INTERVAL seconds.

A host sends Router Solicitations to the All-Routers multicast address. The IP source address is set to either one of the interface's unicast addresses or the unspecified address. The Source Link-Layer Address option SHOULD be set to the host's link-layer address, if the IP source address is not the unspecified address.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

1. Reboot the HUT.
2. Observe the packets transmitted by the HUT.

Observable Results:

> **Step 2:** The HUT should transmit up to MAX_RTR_SOLICITATIONS (3) Router Solicitations RTR_SOLICITATION_INTERVAL (4) seconds apart. The Router Solicitations should be sent from either the link-local address of the HUT or the unspecified address. The destination address should be the All-Routers multicast address. The Router Solicitations may or may not include a Source Link-layer Address option.

Possible Problems:

- None.

Test v6LC.2.2.2: Router Solicitations, Solicited Router Advertisement (Hosts Only)

Purpose: Verify that a host sends valid Router Solicitations at the appropriate time.

References:

- [ND] – Section 6.3.7

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Once the host sends a Router Solicitation, and receives a valid Router Advertisement with a non-zero Router Lifetime, the host MUST desist from sending additional solicitations on that interface, until the next time one of the above events occurs. Moreover, a host SHOULD send at least one solicitation in the case where an advertisement is received prior to having sent a solicitation. Unsolicited Router Advertisements may be incomplete (see Section 6.2.3); solicited advertisements are expected to contain complete information.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Advertisement A

IPv6 Header Next Header: 58 Hop Limit: [See below] Source Address: [See below] Destination Address: All-Node Multicast address
Router Advertisement ICMP Code: [See below] ICMP Checksum: [See below] Source Link-layer Address Option: [See below]

Procedure:

Part A: Valid Router Advertisement, No Source Link-layer Address Option

1. Reboot the HUT.
2. Wait until the HUT transmits a Router Solicitation.
3. TR1 transmits Router Advertisement A without a Source Link-layer Address Option. The Source Address is the link-local address of TR1. The Hop Limit is 255. The ICMP Code is 0. The ICMP Checksum is valid.
4. Wait RTR_SOLICITATION_INTERVAL+MAX_RTR_SOLICITATION_DELAY
5. Observe packets transmitted from the HUT.

Part B: Valid Router Advertisement, Source Link-layer Address Option

6. Repeat Steps 1 through 5 from Part A. Router Advertisement A has a Source Link-layer Address

option.

Part C: Invalid Router Advertisement, Global Source Address

7. Repeat Steps 1 through 5 from Part A. Router Advertisement A has a Source Address of the global address of TR1, but is valid otherwise.

Part D: Invalid Router Advertisement, Bad Hop Limit

8. Repeat Steps 1 through 5 from Part A. Router Advertisement A has a Hop Limit of 2, but is valid otherwise.

Part E: Invalid Router Advertisement, Bad ICMP Checksum

9. Repeat Steps 1 through 5 from Part A. Router Advertisement A has an invalid ICMP checksum, but is otherwise valid.

Part F: Invalid Router Advertisement, Bad ICMP Code

10. Repeat Steps 1 through 5 from Part A. Router Advertisement A has an ICMP Code of 1, but is otherwise valid.

Observable Results:

- *Parts A-B*
  The HUT should transmit only one Router Solicitation. The Router Solicitation should be sent from either the link-local address of the HUT or the unspecified address. The destination address should be the All-Routers multicast address. The Router Solicitation may or may not include a Source Link-layer Address option.
- *Parts C-F*
  The HUT should ignore the invalid Router Advertisement and continue to transmit Router Solicitations. The Router Solicitations should be sent from either the link-local address of the HUT or the unspecified address. The destination address should be the All-Routers multicast address. The Router Solicitations may or may not include a Source Link-layer Address option.

Possible Problems:

- None.

Test v6LC.2.2.3: Host Ignores Router Solicitations (Hosts Only)

Purpose: Verify that a host ignores Router Solicitations and does not update its Neighbor Cache.

References:

- [ND] – Section 6.2.6

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host MUST silently discard any received Router Solicitation messages.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Solicitation A

IPv6 Header Next Header: 58 Destination Address: [See below]
Router Solicitation Source Link-layer Address Option

Procedure:

Part A: All-Router Multicast Destination

1. TN1 transmits Router Solicitation A. The Destination Address is the All-Router multicast Address.
2. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
3. TN1 transmits a link-local Echo Request to the HUT.
4. Wait 2 seconds.
5. Observe the packets transmitted by the HUT.

Part B: All-Nodes Multicast Destination

6. Repeat Steps 1 through 5 from Part A. Router Solicitation A has a Destination Address of the All-Nodes multicast address.

Part C: Link-local Unicast Destination

7. Repeat Steps 1 through 5 from Part A. Router Solicitation A has a Destination Address of the link-local address of the HUT.

Observable Results:

- *Parts A-C*

  In all Parts, the HUT should send a multicast Neighbor Solicitation for TN1 in Step 5, indicating the HUT did not process the Router Solicitation from TN1.

Possible Problems:

- None.

Test v6LC.2.2.4: Router Ignores Invalid Router Solicitations (Routers Only)

Purpose: Verify that a router ignores invalid Router Solicitations.

References:

- [ND] – Section 6.1.1, 6.2.6

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A router MUST silently discard any received Router Solicitation messages that do not satisfy all of the following validity checks:

- The IP Hop Limit field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.

- If the message includes an IP Authentication Header, the message authenticates correctly.

- ICMP Checksum is valid.

- ICMP Code is 0.

- ICMP length (derived from the IP length) is 8 or more octets.

- All included options have a length that is greater than zero.

- If the IP source address is the unspecified address, there is no source link-layer address option in the message.

The contents of the Reserved field, and of any unrecognized options, MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new options; backward-incompatible changes may use different Code values.

The contents of any defined options that are not specified to be used with Router Solicitation messages MUST be ignored and the packet processed as normal. The only defined option that may appear is the Source Link-Layer Address option.

A solicitation that passes the validity checks is called a "valid solicitation".

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

Part A: Hop Limit is not 255

1. TN1 transmits a Router Solicitation with an IPv6 Hop Limit of 254. The Router Solicitation is valid otherwise.
2. Observe the packets transmitted by the RUT.

Part B: ICMPv6 checksum is not valid

3. TN1 transmits a Router Solicitation with an invalid ICMPv6 checksum. The Router Solicitation is valid otherwise.
4. Observe the packets transmitted by the RUT.

Part C: ICMPv6 code is not 0

5. TN1 transmits a Router Solicitation with an invalid ICMPv6 code of 1. The Router Solicitation is valid otherwise.
6. Observe the packets transmitted by the RUT.

Part D: ICMPv6 length is less than 8 Octets

7. TN1 transmits a Router Solicitation with an ICMPv6 length of 6. The Router Solicitation is valid otherwise.
8. Observe the packets transmitted by the RUT.

Part E: Option has length 0

9. TN1 transmits a Router Solicitation that contains an Option with length 0. The Router Solicitation is valid otherwise.
10. Observe the packets transmitted by the RUT.

Part F: Unspecified IP source address and a source link-layer address option

11. TN1 transmits a Router Solicitation with an unspecified IP source address and a source link-layer address option. The Router Solicitation is valid otherwise.
12. Observe the packet transmitted by the RUT.

Observable Results:

- *Parts A-F*
  In all Parts, the RUT must discard the Router Solicitation from TN1 and must not transmit a corresponding Router Advertisement within MAX_RA_DELAY_TIME (0.5) seconds.

Possible Problems:

- None.

Test v6LC.2.2.5: Router Sends Valid Router Advertisement (Routers Only)

Purpose: Verify that a router sends valid Router Advertisements.

References:

- [ND] – Section 6.1.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node MUST silently discard any received Router Advertisement messages that do not satisfy all of the following validity checks:

- IP Source Address is a link-local address. Routers must use their link-local address as the source for Router Advertisement and Redirect messages so that hosts can uniquely identify routers.

- The IP Hop Limit field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.

- If the message includes an IP Authentication Header, the message authenticates correctly.

- ICMP Checksum is valid.

- ICMP Code is 0.

- ICMP length (derived from the IP length) is 16 or more octets.

- All included options have a length that is greater than zero.

The contents of the Reserved field, and of any unrecognized options, MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new options; backward-incompatible changes may use different Code values.

The contents of any defined options that are not specified to be used with Router Advertisement messages MUST be ignored and the packet processed as normal. The only defined options that may appear are the Source Link-Layer Address, Prefix Information and MTU options.

An advertisement that passes the validity checks is called a "valid advertisement".

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

1. TN1 transmits a valid Router Solicitation.
2. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT must transmit valid Router Advertisements that satisfy all of the following validity checks:

- IP Source Address is a link-local address.
- The IP Hop Limit field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.
- If the message includes an IP Authentication Header, the message authenticates correctly.
- ICMP Checksum is valid.
- ICMP Code is 0.
- ICMP length (derived from the IP length) is 16 or more octets.
- All included options have a length that is greater than zero.

Possible Problems:

- None.

Test v6LC.2.2.6: Router Does Not Send Router Advertisements on Non-advertising Interface (Routers Only)

Purpose: Verify that a router does not send Router Advertisements on non-advertising interfaces.

References:

- [ND] – Section 6.2.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The term "advertising interface" refers to any functioning and enabled multicast interface that has at least one unicast IP address assigned to it and whose corresponding AdvSendAdvertisements flag is TRUE. A router MUST NOT send Router Advertisements out any interface that is not an advertising interface.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

Part A: No advertising interfaces

1. Configure Interface A on the RUT to be a non-advertising interface.
2. Configure TR1 to transmit a RS to the RUT on Interface A.
3. Observe the packets transmitted by the RUT on Interface A.

Part B: Advertising interface

4. If the RUT supports two network interfaces. Configure Interface A on the RUT to be an advertising interface and interface B to be a non-advertising interface.
5. Configure TR1 to transmit a RS to the RUT on Interface A and on Interface B.
6. Observe the packets transmitted by the RUT on Interface A and Interface B.

Observable Results:

- *Part A*

  Step 3: The RUT must not send Router Advertisements out on Interface A.
- *Part B*

  Step 4: The RUT must send Router Advertisements out on Interface A. The RUT must not send Router Advertisements out on Interface B.

Possible Problems:

- None.

Test v6LC.2.2.7: Sending Unsolicited Router Advertisements (Routers Only)

Purpose: Verify that a router sends the first few advertisements (up to MAX_INITIAL_RTR_ADVERTISEMENTS) from an interface when it becomes an advertising interface at an maximum interval value of MAX_INITIAL_RTR_ADVERT_INTERVAL (16) seconds.

References:

- [ND] – Section 6.2.4, 6.2.6

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Unsolicited Router Advertisements are not strictly periodic: the interval between subsequent transmissions is randomized to reduce the probability of synchronization with the advertisements from other routers on the same link [SYNC]. Each advertising interface has its own timer. Whenever a multicast advertisement is sent from an interface, the timer is reset to a uniformly-distributed random value between the interface's configured MinRtrAdvInterval and MaxRtrAdvInterval; expiration of the timer causes the next advertisement to be sent and a new random value to be chosen.

For the first few advertisements (up to MAX_INITIAL_RTR_ADVERTISEMENTS) sent from an interface when it becomes an advertising interface, if the randomly chosen interval is greater than MAX_INITIAL_RTR_ADVERT_INTERVAL, the timer SHOULD be set to MAX_INITIAL_RTR_ADVERT_INTERVAL instead. Using a smaller interval for the initial advertisements increases the likelihood of a router being discovered quickly when it first becomes available, in the presence of possible packet loss.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

1. Configure Interface A on the RUT to be an advertising interface with a MinRtrAdvInterval of 5 seconds and a MaxRtrInterval of 10 seconds.
2. Observe the packets transmitted by the RUT on Interface A.

Observable Results:

Step 2: The RUT should transmit the first MAX_INITIAL_RTR_ADVERTISEMENTS (3) at randomly chosen intervals of no greater than MAX_INITIAL_RTR_ADVERT_INTERVAL (16) seconds. The RUT should transmit the following consecutive Router Advertisements at randomly chosen intervals between the interface’s configured MinRtrAdvInterval (5) and MaxRtrAdvInterval (10) seconds, and it MUST NOT transmit Router Advertisements more frequently than indicated by MinRtrAdvInterval (5) seconds.

Possible Problems:

- None.

Test v6LC.2.2.8: Ceasing to Be An Advertising Interface (Routers Only)

Purpose: Verify that a router sends correct Router Advertisements when its interface ceases to be an advertising interface.

References:

- [ND] – Section 6.2.5

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: An interface may cease to be an advertising interface, through actions of system management such as:

- changing the AdvSendAdvertisements flag of an enabled interface from TRUE to FALSE, or

- administratively disabling the interface, or

- shutting down the system.

In such cases the router SHOULD transmit one or more (but not more than MAX_FINAL_RTR_ADVERTISEMENTS) final multicast Router Advertisements on the interface with a Router Lifetime field of zero. In the case of a router becoming a host, the system SHOULD also depart from the all-routers IP multicast group on all interfaces on which the router supports IP multicast (whether or not they had been advertising interfaces). In addition, the host MUST insure that subsequent Neighbor Advertisement messages sent from the interface have the Router flag set to zero.

Note that system management may disable a router's IP forwarding capability (i.e., changing the system from being a router to being a host), a step that does not necessarily imply that the router's interfaces stop being advertising interfaces. In such cases, subsequent Router Advertisements MUST set the Router Lifetime field to zero.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Procedure:

1. Configure Interface A on the RUT to be an advertising interface.
2. Configure Interface A on the RUT to discontinue be an advertising interface.
3. Observe the packets transmitted by the RUT on Interface A.

Observable Results:

Step 3: The RUT should transmit no more than MAX_FINAL_RTR_ADVERTISEMENTS (3) final multicast Router Advertisement on the interface with a Router Lifetime field of zero.

Possible Problems:

- None.

Test v6LC.2.2.9: Processing Router Solicitations (Routers Only)

Purpose: Verify that a router correctly processes Router Solicitations and transmits Router Advertisements.

References:

- [ND] – Section 6.2.6

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: In addition to sending periodic, unsolicited advertisements, a router sends advertisements in response to valid solicitations received on an advertising interface. A router MAY choose to unicast the response directly to the soliciting host's address (if the solicitation's source address is not the unspecified address), but the usual case is to multicast the response to the all-nodes group. In the latter case, the interface's interval timer is reset to a new random value, as if an unsolicited advertisement had just been sent (see Section 6.2.4).

In all cases, Router Advertisements sent in response to a Router Solicitation MUST be delayed by a random time between 0 and MAX_RA_DELAY_TIME seconds. (If a single advertisement is sent in response to multiple solicitations, the delay is relative to the first solicitation.) In addition, consecutive Router Advertisements sent to the all-nodes multicast address MUST be rate limited to no more than one advertisement every MIN_DELAY_BETWEEN_RAS seconds.

A router might process Router Solicitations as follows:

- Upon receipt of a Router Solicitation, compute a random delay within the range 0 through MAX_RA_DELAY_TIME. If the computed value corresponds to a time later than the time the next multicast Router Advertisement is scheduled to be sent, ignore the random delay and send the advertisement at the already-scheduled time.

- If the router sent a multicast Router Advertisement (solicited or unsolicited) within the last MIN_DELAY_BETWEEN_RAS seconds, schedule the advertisement to be sent at a time corresponding to MIN_DELAY_BETWEEN_RAS plus the random value after the previous advertisement was sent. This ensures that the multicast Router Advertisements are rate limited.

- Otherwise, schedule the sending of a Router Advertisement at the time given by the random value.

Note that a router is permitted to send multicast Router Advertisements more frequently than indicated by the MinRtrAdvInterval configuration variable so long as the more frequent advertisements are responses to Router Solicitations. In all cases, however, unsolicited multicast advertisements MUST NOT be sent more frequently than indicated by MinRtrAdvInterval.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Solicitation A	Router Solicitation B
IPv6 Header Next Header: 58 Source Address: TN1's Link Local Address	IPv6 Header Next Header: 58 Source Address: Unspecified Address
Router Solicitation	Router Solicitation

Procedure:

Part A: MAX_RA_DELAY_TIME

1. TN1 transmits Router Solicitation A twice, 3 seconds apart. The Destination Address is the all-routers multicast address.
2. Observe the packets transmitted by the RUT.

Part B: MIN_DELAY_BETWEEN_RAS

3. Configure the RUT with a MinRtrAdvInterval of 30 seconds and a MaxRtrAdvInterval of 40 seconds.
4. TN1 transmits Router Solicitation B twice, 2 seconds apart. The destination Address is the all-routers multicast address.
5. Observe the packets transmitted by the RUT. Repeat Step 4.

Observable Results:

- *Part A*

  Step 2: The RUT MUST transmit a Router Advertisement between 0 and MAX_RA_DELAY_TIME (0.5) seconds after the receipt of each Router Solicitation A.
- *Part B*

  Step 5: The RUT MUST NOT transmit more than one advertisement every MIN_DELAY_BETWEEN_RAS (3) seconds.

Possible Problems:

- None.

Test v6LC.2.2.10: Default Router Switch (Hosts Only)

Purpose: Verify that a host maintains at least two routers in its Default Router List and will switch routers when the router in use fails.

References:

- [ND] – Sections 5.2, 5.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host should retain entries in the Default Router List and the Prefix List until their lifetimes expire. However, a host may delete old entries prematurely if it is low on memory. If not all routers are kept on the Default Router list, a host should retain at least two entries in the Default Router List (preferably more) in order to maintain robust connectivity for off-link destinations. For efficiency reasons, next-hop determination is not performed on every packet that is sent. Instead, the results of next-hop determination computations are saved in the Destination Cache. When the sending host has a packet to send, it first examines the Destination Cache. If no entry exists for the destination, next-hop determination is invoked to create a Destination Cache entry.

Next-hop determination is done the first time traffic is sent to a destination. As long as subsequent communication to that destination proceeds successfully, the Destination Cache entry continues to be used. If at some point communication ceases to proceed, as determined by the Neighbor Unreachability Detection algorithm, next-hop determination may need to be performed again. For example, traffic through a failed router should be switched to a working router.

Test Setup: No Common Test Setup is performed. The common cleanup procedure from Common Test Setup 1.1 is performed after each part.

Router Advertisement A	Router Advertisement B
IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address	IPv6 Header Next Header: 58 Source Address: TR2's Link Local Address
Router Advertisement Router Lifetime: 45 seconds Reachable Time: 10 seconds Retransmit Interval: 1 second Prefix Length: 64 L Bit: 1 (on-link) Prefix: TN1's Global Prefix	Router Advertisement Router Lifetime: 45 seconds Reachable Time: 10 seconds Retransmit Interval: 1 second Prefix Length: 64 L Bit: 1 (on-link) Prefix: TN1's Global Prefix

Packet A

IPv6 Header Next Header: 58 Source Address: TN2's Global Address Destination Address: HUT's Global Address
ICMPv6 Echo Request

Procedure:

1. TR1 transmits Router Advertisement A.
2. TR1 transmits Packet A, an Echo Request.
3. Observe the packets transmitted by the HUT. TR1 transmits a Neighbor Advertisement in response to any Neighbor Solicitations from the HUT.
4. TR2 transmits Router Advertisement B.
5. TN2 transmits Packet A every 3 seconds for 30 seconds. Packet A is an ICMPv6 Echo Request that has an off-link global source address.
6. Observe the packets transmitted by the HUT.
7. When Reachable Time expires, and the HUT solicits TR1, no Neighbor Advertisements are transmitted by TR1.
8. Observe the packets transmitted by the HUT.

Observable Results:

Steps 3: The HUT should transmit a Neighbor Solicitation with a Target Address equal to TR1's link-local address. The HUT should send an Echo Reply to TN2 via TR1 in response to Packet A.
Step 6: The HUT should send Echo Replies to TR1's link local address until Reachable Time expires. When Reachable Time expires, the HUT should send 3 Neighbor Solicitations to TR1's link local address.
Step 8: The HUT selects TR2 from its Default Router list. The HUT sends Neighbor Solicitations to TR2's link local address. After sending the packets to TR2, the HUT probes TR2 as a side effect.

Possible Problems:

- None.

Test v6LC.2.2.11: Router Advertisement Processing, Validity (Hosts Only)

Purpose: Verify that a host properly discards an invalid Router Advertisement.

References:

- [ND] – Section 6.1.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node MUST silently discard any received Router Advertisement messages that do not satisfy all of the following validity checks:

- IP Source Address is a link-local address. Routers must use their link-local address as the source for Router Advertisement and Redirect messages so that hosts can uniquely identify routers.
- The IP Hop Limit field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.
- If the message includes an IP Authentication Header, the message authenticates correctly.
- ICMP Checksum is valid.
- ICMP Code is 0.
- ICMP length (derived from the IP length) is 16 or more octets.
- All included options have a length that is greater than zero.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Router Advertisement

IPv6 Header Next Header: 58 Hop Limit: [See below] Source Address: [See below] Destination Address: Multicast Address
Router Advertisement ICMP Code: [See below] ICMP Checksum: [See below] Router Lifetime: 20 seconds Reachable Time: 600 seconds Retransmit Interval: 1 second
Source Link-layer Address Option

Procedure:

Part A: Global Source Address

1. TR1 transmits the Router Advertisement. The Source Address is the global address of TR1. The Router Advertisements is valid otherwise.
2. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
3. TR1 transmits a link-local Echo Request to the HUT.
4. Wait 2 seconds and observe the packets transmitted by the HUT.

Part B: Hop Limit less than 255

5. TR1 transmits the Router Advertisement. The Hop Limit is 2. The Router Advertisement is valid otherwise.
6. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
7. TR1 transmits a link-local Echo Request to the HUT.
8. Wait 2 seconds and observe the packets transmitted by the HUT.

Part C: Invalid Checksum

9. TR1 transmits the Router Advertisement. The ICMP Checksum is invalid. The Router Advertisement is valid otherwise.
10. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
11. TR1 transmits a link-local Echo Request to the HUT.
12. Wait 2 seconds and observe the packets transmitted by the HUT.

Part D: Invalid ICMP Code

13. TR1 transmits the Router Advertisement. The ICMP Code is 1. The Router Advertisement is valid otherwise.
14. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
15. TR1 transmits a link-local Echo Request to the HUT.
16. Wait 2 seconds and observe the packets transmitted by the HUT.

Part E: Invalid ICMP Length

17. TR1 transmits the Router Advertisement with an ICMP length of 14. The Router Advertisement is valid otherwise.
18. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
19. TR1 transmits a link-local Echo Request to the HUT.
20. Wait 2 seconds and observe the packets transmitted by the HUT.

Part F: Option of Length 0

21. TR1 transmits the Router Advertisement with an option of length 0. The Router Advertisement is valid otherwise.
22. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
23. TR1 transmits a link-local Echo Request to the HUT.
24. Wait 2 seconds and observe the packets transmitted by the HUT.

Observable Results:

- *Parts A-F*

  In all parts, the HUT should transmit a multicast Neighbor Solicitation for TR1, indicating the HUT did not have a NCE for TR1.

Possible Problems:

- None.

Test v6LC.2.2.12: Router Advertisement Processing, Cur Hop Limit

Purpose: Verify that a node properly processes the Cur Hop Limit field of a Router Advertisement.

References:

- [ND] – Sections 4.2, 6.2.1 and 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Cur Hop Limit is the default value that should be placed in the Hop Count field of the IP header for outgoing IP packets. A value of zero means unspecified (by this router.)

If the received Cur Hop Limit value is non-zero the host SHOULD set its CurHopLimit variable to the received value.

Some Router Advertisement fields (e.g., Cur Hop Limit, Reachable Time and Retrans Timer) may contain a value denoting unspecified. In such cases, the parameter should be ignored and the host should continue using whatever value it is already using.

Some host variables (e.g., CurHopLimit, RetransTimer, and ReachableTime) apply to all nodes including routers. In practice, these variables may not actually be present in routers, since their contents can be derived from the router configuration variables. However, external router behavior MUST be the same as host behavior with respect to these variables.

Test Setup: Common Test Setup 1.1 is performed. For Part B, Common Test Setup 1.3 is performed. The common cleanup procedure is performed after each part.

Procedure:

Part A: Unspecified

1. TN1 transmits an Echo Request to the NUT.
2. Observe the packets transmitted by the NUT.
3. If the NUT is a host, TR1 transmits a Router Advertisement with a Cur Hop Limit value of 0 (Zero). If the NUT is a router, configure the Cur Hop Limit to a value of 0 (zero).
4. TN1 transmits an Echo Request to the NUT.
5. Observe the packets transmitted by the NUT.

Part B: Non-Zero

6. TN1 transmits an Echo Request to the NUT.
7. Observe the packets transmitted by the NUT.
8. If the NUT is a host, TR1 transmits a Router Advertisement with a Cur Hop Limit value of 15. If the NUT is a router, configure the Cur Hop Limit to a value of 15.
9. TN1 transmits an Echo Request to the NUT.

10. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT should respond to the Request from TN1. Observe the Hop Limit value in the Echo Reply packet the NUT transmits.
  Step 5: The NUT should respond to the Request from TN1. The Hop Limit value in the Echo Reply should be the same as was used in step 2.
- *Part B*

  Step 7: The NUT should respond to the Request from TN1. Observe the Hop Limit value in the Echo Reply packet the NUT transmits.
  Step 10: The NUT should respond to the Request from TN1. The Hop Limit value in the Echo Reply should be 15.

Possible Problems:

- None.

Test v6LC.2.2.13: Router Advertisement Processing, Router Lifetime (Hosts Only)

Purpose: Verify that a host properly processes a Router Advertisement and the Router Lifetime field within it.

References:

- [ND] – Section 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: On receipt of a valid Router Advertisement, a host extracts the source address of the packet and does the following:

- If the address is not already present in the host's Default Router List, and the advertisement's Router Lifetime is non-zero, create a new entry in the list, and initialize its invalidation timer value from the advertisement's Router Lifetime field.
- If the address is already present in the host's Default Router List as a result of a previously received advertisement, reset its invalidation timer to the Router Lifetime value in the newly-received advertisement.
- If the address is already present in the host's Default Router List and the received Router Lifetime value is zero, immediately timeout the entry as specified in Section 6.3.5.

Test Setup: For Part A, Common Test Setup 1.1 is performed. For Parts B and C, Common Test Setup 1.3 is performed. The common cleanup procedure is performed after each part.

Router Advertisement
IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: All-Nodes Multicast Address
Router Advertisement Router Lifetime: 20 seconds Reachable Time: 600 seconds Retransmit Interval: 1 second
Prefix Option Valid Lifetime: 100 seconds Preferred Lifetime: 20 seconds Prefix: TR1's Global Prefix

Procedure:

Part A: Router Lifetime Updated with Same Lifetime

1. TR1 transmits the Router Advertisement.
2. TN2 transmits a global Echo Request to the HUT every second for 19 seconds.
3. Observe the packets transmitted by the HUT.
4. TR1 transmits the Router Advertisement.
5. TN2 transmits a global Echo Request to the HUT every second for 21 seconds.
6. Observe the packets transmitted by the HUT.

Part B: Router Lifetime Set to Zero

7. TN2 transmits a global Echo Request to the HUT.
8. Observe the packets transmitted by the HUT.
9. TR1 transmits a Router Advertisement with Router Lifetime set to zero.
10. TN2 transmits a global Echo Request to the HUT.
11. Observe the packets transmitted by the HUT.
12. TR2 transmits a Router Advertisement with Router Lifetime set to zero.
13. TN2 transmits a global Echo Request to the HUT.
14. Observe the packets transmitted by the HUT.
15. TR3 transmits a Router Advertisement with Router Lifetime set to zero.
16. TN2 transmits a global Echo Request to the HUT.
17. Observe the packets transmitted by the HUT.

Part C: Router Lifetime Set to Five; Allowed to Expire

18. TN2 transmits a global Echo Request to the HUT.
19. Observe the packets transmitted by the HUT.
20. TR1 transmits a Router Advertisement with Router Lifetime set to five.
21. Wait seven seconds.
22. TN2 transmits a global Echo Request to the HUT.
23. Observe the packets transmitted by the HUT.
24. TR2 transmits a Router Advertisement with Router Lifetime set to five.
25. Wait seven seconds.
26. TN2 transmits a global Echo Request to the HUT.
27. Observe the packets transmitted by the HUT.
28. TR3 transmits a Router Advertisement with Router Lifetime set to five.
29. Wait seven seconds.
30. TN2 transmits a global Echo Request to the HUT.
31. Observe the packets transmitted by the HUT.

Observable Results:

- *Part A*

  Step 3: The HUT should respond to the Echo Requests from TN2 using TR1 as a first hop.
  Step 4: The HUT should update the Router Lifetime for TR1.
  Step 6: The HUT should respond to the Echo Requests from TN2 using TR1 as a first hop until the Router Lifetime expires. In response to the final Echo Request, the HUT MUST not transmit an Echo Reply or transmit multicast NS's with a target address set to TR1's link-local address.
- *Part B*

Step 8: The HUT should use TR1, TR2, or TR3 as a first hop.
Step 11: The HUT should use TR2 or TR3 as a first hop.
Step 14: The HUT should use TR3 as a first hop.
Step 17: The HUT MUST not transmit an Echo Reply or transmit multicast NS's with a target address set to TR1's link-local address.

- *Part C*

  Step 19: The HUT should use TR1, TR2, or TR3 as a first hop.
  Step 23: The HUT should use TR2 or TR3 as a first hop.
  Step 27: The HUT should use TR3 as a first hop.
  Step 31: The HUT MUST not transmit an Echo Reply or transmit multicast NS's with a target address set to TR1's link-local address.

Possible Problems:

- None.

Test v6LC.2.2.14: Router Advertisement Processing, Reachable Time

Purpose: Verify that a node updates its BaseReachableTime variable and re-computes its ReachableTime variable upon receipt of a Router Advertisement or a configuration with a specified Reachable Time.

References:

- [ND] – Sections 6.2.1 and 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If the received Reachable Time value is non-zero, the host SHOULD set its BaseReachableTime variable to the received value. If the new value differs from the previous value, the host SHOULD re-compute a new random ReachableTime value. ReachableTime is computed as a uniformly distributed random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR times the BaseReachableTime. Using a random component eliminates the possibility of Neighbor Unreachability Detection messages synchronizing with each other.

Some host variables (e.g., CurHopLimit, RetransTimer, and ReachableTime) apply to all nodes including routers. In practice, these variables may not actually be present in routers, since their contents can be derived from the router configuration variables. However, external router behavior MUST be the same as host behavior with respect to these variables. In particular, this includes the occasional randomization of the ReachableTime value as described in Section 6.3.2.

Test Setup: No Common Test Setup is performed. The common cleanup procedure is performed after each part.

Router Advertisement

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address
Router Advertisement Router Lifetime: [see below] Reachable Time: [see below] Retransmit Interval: 1 second

Procedure:

Part A: RA Processing – Reachable Time (Host Only)

1. TR1 transmits the Router Advertisement with a Router Lifetime of 0 seconds and a Reachable Time of 10 seconds.
2. TN1 transmits a link-local Echo Request to the HUT. TN1 must reply to any Neighbor

Solicitations from the HUT.

3. Observe the packets transmitted by the HUT.
4. Repeat Step 2 every second for 40 seconds.
5. Observe the packets transmitted by the HUT.
6. TR1 transmits the Router Advertisement with a Reachable Time of 40 seconds.
7. Repeat Step 2 every seconds for 140 seconds.
8. Observe the packets transmitted by the HUT.

Part B: Reachable Time Configuration (Routers Only)

9. Configure the RUT to transmit Router Advertisements with a Router Lifetime value of 0 seconds and a Reachable Time of 10 seconds.
10. TN1 transmits a link-local Echo Request to the RUT. TN1 must reply to any Neighbor Solicitations from the RUT.
11. Observe the packets transmitted by the RUT.
12. Repeat Step 10 every second for 40 seconds.
13. Observe the packets transmitted by the RUT

Observable Results:

- *Part A*

  Step 3: The HUT should solicit for TN1's link-local address and transmit an Echo Reply.
  Step 5: The HUT should transmit a Neighbor Solicitation with a Target Address of TN1's link-local address at an interval between 5 and 15 seconds.
  Step 8: The HUT should transmit Neighbor Solicitations at an interval between 20 and 60 seconds.
- *Part B*

  Step 11: The RUT should solicit for TN1's link-local address and transmit an Echo Reply.
  Step 13: The RUT should transmit a Neighbor Solicitation with a Target Address of TN1's link-local address at an interval between 5 and 20 seconds.

Possible Problems:

- None.

Test v6LC.2.2.15: Router Advertisement Processing, Neighbor Cache (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache upon receipt of a Router Advertisement.

References:

- [ND] – Sections 6.3.4 and 7.3.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: After extracting information from the fixed part of the Router Advertisement message, the advertisement is scanned for valid options. If the advertisement contains a Source Link-Layer Address option, the link-layer address SHOULD be recorded in the Neighbor Cache entry for the router (creating an entry if necessary) and the IsRouter flag in the Neighbor Cache entry MUST be set to TRUE. If no Source Link-Layer option is included, but a corresponding Neighbor Cache entry exists, its IsRouter flag MUST be set to TRUE. The IsRouter flag is used by Neighbor Unreachability Detection to determine when a router changes to being a host (i.e., no longer capable of forwarding packets). If a Neighbor Cache entry is created for the router, its reachability state MUST be set to STALE as specified in Section 7.3.3. If a cache entry already exists and is updated with a different link-layer address, its reachability state MUST also be set to STALE.

Test Setup: No Common Test Setup is performed. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The common cleanup procedure is performed after each part.

Router Advertisement A
IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address
Router Advertisement Router Lifetime: 0 seconds Reachable Time: 10 seconds Retransmit Interval: 1 second Source Link-layer Option

Echo Request B

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: HUT's Link-local Address
ICMPv6 Echo Request

Neighbor Advertisement C

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: HUT's Link-local Address
Neighbor Advertisement Router flag: 0 Solicited flag: 1 Override flag: 1 Target Address: TR1's Link-local Address

Procedure:

Part A: RA processing, no NCE

1. TR1 transmits Router Advertisement A.
2. TR1 transmits an Echo Request to the HUT.
3. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Part B: RA processing, NCE INCOMPLETE

4. TR1 transmits Echo Request B. TR1 does not respond to any Neighbor Solicitations from the HUT.
5. Observe the packets transmitted by the HUT and check the NCE of TR1 on the HUT.
6. TR1 transmits Router Advertisement A.
7. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Part C: RA with SLLA changed, NCE REACHABLE

8. TR1 transmits Echo Request B. TR1 does not respond to any Neighbor Solicitations from the HUT.
9. Observe the packets transmitted by the HUT and check the NCE of TR1 on the HUT.
10. TR1 transmits Neighbor Advertisement C.
11. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.
12. TR1 transmits Router Advertisement A with a different Source Link-layer Address.
13. TR1 transmits an Echo Request to the HUT.
14. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Part D: RA with SLLA unchanged, NCE REACHABLE

15. Repeat Steps 8 through 14, transmitting Router Advertisement A in Step 12 with the same Source Link-layer Address.
16. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Part E: RA with SLLA changed, NCE PROBE

17. TR1 transmits Echo Request B. TR1 does not respond to any Neighbor Solicitations from the HUT.
18. Observe the packets transmitted by the HUT and check the NCE of TR1 on the HUT.
19. TR1 transmits Neighbor Advertisement C.
20. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.
21. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR) seconds.
22. TR1 transmits Echo Request B.
23. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.
24. Wait (DELAY_FIRST_PROBE_TIME) seconds.
25. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.
26. TR1 transmits Router Advertisement A with a different Source Link-layer Address.
27. TR1 transmits an Echo Request to the HUT.
28. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Part F: RA with SLLA unchanged, NCE PROBE

29. Repeat Steps 17 through 28, transmitting Router Advertisement A in Step 26 with the same Source Link-layer Address.
30. Check the NCE of TR1 on the HUT and observe the packets transmitted by the HUT.

Observable Results:

- *Part A*

  Step 3: The HUT must create a NCE for TR1, set the NCE's state to **STALE**, and record TR1's Link-layer Address. Because the HUT's NCE for TR1 is in state **STALE**, the HUT should send an Echo Reply to TR1 and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the HUT should send a unicast Neighbor Solicitation to TR1.
- *Part B*

  Step 5: The HUT should create a Neighbor Cache Entry for TR1 and set the state of the Entry to **INCOMPLETE**. The HUT should send a multicast Neighbor Solicitation to TR1.
  Step 7: The HUT must update the state of TR1's NCE to **STALE** and update its Link-layer Address. Because the HUT's NCE for TR1 is in state **STALE**, the HUT should send an Echo Reply to TR1's earlier request using the received Link-Layer address and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the HUT should send a unicast Neighbor Solicitation to TR1.
- *Part C*

  Step 9: The HUT should create a Neighbor Cache Entry for TR1 and set the state of the Entry to **INCOMPLETE**. The HUT should send a multicast Neighbor Solicitation to TR1.
  Step 11: The HUT should update the state of TR1's NCE to **REACHABLE** and record TR1's Link-layer Address. Because the HUT is in state **REACHABLE**, after receiving the earlier Echo Request from TR1, the HUT should send an Echo Reply using the received Link-Layer Address. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR1.

Step 14: The HUT must change the state of the TR1's NCE to STALE and update its Link-layer Address according to the Router Advertisement received in Step 12. Because the HUT's NCE for TR1 is in state **STALE**, the HUT should send an Echo Reply to TR1 using the new Link-Layer address and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the HUT should send a unicast Neighbor Solicitation to TR1.

- *Part D*

  Step 9: The HUT should create a Neighbor Cache Entry for TR1 and set the state of the Entry to **INCOMPLETE**. The HUT should send a multicast Neighbor Solicitation to TR1.
  Step 11: The HUT should update the state of TR1's NCE to **REACHABLE** and record TR1's Link-layer Address. Because the HUT is in state **REACHABLE**, after receiving the earlier Echo Request from TR1, the HUT should send an Echo Reply using the received Link-Layer Address. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR1.
  Step 14: The HUT must not change the state of the TN1's NCE. After receiving the Echo Request from TR1, the HUT should send an Echo Reply using the same Link-Layer Address. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR1.

- *Part E*

  Step 18: The HUT should create a Neighbor Cache Entry for TR1 and set the state of the Entry to **INCOMPLETE**. The HUT should send a multicast Neighbor Solicitation to TR1.
  Step 20: The HUT should update the state of TR1's NCE to **REACHABLE** and record TR1's Link-layer Address. Because the HUT is in state **REACHABLE**, after receiving the earlier Echo Request from TR1, the HUT should send an Echo Reply using the received Link-Layer Address. After DELAY_FIRST_PROBE_TIME, the NUT should not send a unicast Neighbor Solicitation to TR1.
  Step 23: The HUT should update the state of TR1's NCE to STALE, send an Echo Reply to TR1 using the same Link-Layer address and enter state **DELAY**.
  Step 25: The HUT should update the state of TR1's NCE to **PROBE** by sending a unicast Neighbor Solicitation to TR1.
  Step 28: The HUT must change the state of the TN1's NCE to **STALE** and update TN1's Link-Layer Address according to the received Router Advertisement. Because the HUT's NCE for TN1 is in state **STALE**, the HUT should send an Echo Reply to TN1 using the new Link-Layer Address and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the NUT should send a unicast Neighbor Solicitation to TR1.

- *Part F*

  Step 18: The HUT should create a Neighbor Cache Entry for TR1 and set the state of the Entry to **INCOMPLETE**. The HUT should send a multicast Neighbor Solicitation to TR1.
  Step 20: The HUT should update the state of TR1's NCE to **REACHABLE** and record TR1's Link-layer Address. Because the HUT is in state **REACHABLE**, after receiving the earlier Echo Request from TR1, the HUT should send an Echo Reply using the received Link-Layer Address. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR1.
  Step 23: The HUT should update the state of TR1's NCE to STALE, send an Echo Reply to TR1 using the same Link-Layer address and enter state **DELAY**.
  Step 25: The HUT should update the state of TR1's NCE to **PROBE** by sending a unicast Neighbor Solicitation to TR1.
  Step 28: The HUT must not change the state of the TR1's NCE. The HUT must continue to

be in state **PROBE** and send unicast Neighbor Solicitations to TR1 up to MAX_UNICAST_SOLICIT times.

Possible Problems:

- None.

Group 3: Redirect Function

Scope

The following tests cover the Redirect function in IPv6.

Overview

Tests in this group verify that a node properly processes valid, suspicious, and invalid Redirect messages. These tests also verify a node uses the appropriate first hop when redirected twice, receiving invalid options, having no entry in its Destination Cache, or when the new first hop is not reachable. These tests also verify interactions between Target Link-layer Address options with the Neighbor Cache.

Test v6LC.2.3.1: Redirected On-link: Valid (Hosts Only)

Purpose: Verify that a host properly processes valid Redirect messages when redirected on-link.

References:

- [ND] – Sections 4.6.1, 4.6.3, and 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for he destination, an implementation SHOULD create such an entry. Hosts can be redirected to a better first-hop but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMPv6 Target Address equal to the ICMPv6 Destination Address. If the ICMPv6 Target and Destination Addresses are both the same, the host MUST treat the Target as on-link. The Redirected Header option is used in Redirect messages and contains all or part of the packet that is being redirected. The Target Link-layer Address option is used in Redirect and Neighbor Advertisement messages and contains the link-layer address for the target.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

IPv6 Destination Address	TLLA Option	Redirected Packet Option	Part
Link-local (HUT)	No	No	A
Link-local (HUT)	No	Yes	B
Link-local (HUT)	Yes	No	C
Link-local (HUT)	Yes	Yes	D
Global (HUT)	No	No	E
Global (HUT)	No	Yes	F
Global (HUT)	Yes	No	G
Global (HUT)	Yes	Yes	H

Procedure:

Parts A through H: Destination Addresses, TLLA Options, and Redirected Packet Options

1. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the off-link global address of TN1. The Redirect

message contains a Target Link-layer Address option or Redirected Packet option according to the table above.

4. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
5. Observe the packets transmitted by the HUT.
6. Repeat Steps 1 through 5 for each Part B through H, using the Redirect message detailed in the table above in Step 3.

Observable Results:

- *Parts A, B, E, and F*
  Step 2: The HUT should respond to the Echo Request using TR1 as a first hop.
  Step 5: The HUT should transmit a Neighbor Solicitation for TN1's global address and an Echo Reply directly on-link to TN1, indicating the HUT processed the Redirect message.
- *Parts C, D, G, and H*
  Step 2: The HUT should respond to the Echo Request using TR1 as a first hop.
  Step 5: The HUT should transmit an Echo Reply directly on-link to TN1, indicating the HUT processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.2: Redirected On-link: Suspicious (Hosts Only)

Purpose: Verify that a host properly processes suspicious Redirect messages when redirected on-link.

References:

- [ND] – Sections 4.5, 4.6.3, 8.1, and 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for the destination, an implementation SHOULD create such an entry. Hosts can be redirected to a better first-hop but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMPv6 Target Address equal to the ICMPv6 Destination Address. If the ICMPv6 Target and Destination Addresses are both the same, the host MUST treat the Target as on-link. The Redirected Header option is used in Redirect messages and contains all or part of the packet that is being redirected.

The contents of the Reserved field, and any unrecognized options MUST be ignored.

A host MUST NOT consider a redirect invalid just because the Target Address of the redirect is not covered under one of the link's prefixes. Part of the semantics of the Redirect message is that the Target Address is on-link.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Option Unrecognized

1. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the global address of TN1. The Redirect message contains a Target Link-layer Address option. The Redirect message also contains an unrecognized option.
4. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
5. Observe the packets transmitted by the HUT.

Part B: Reserved Field is Non-zero

6. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message has a non-zero Reserved field.

Part C: Target Address not Covered by On-link Prefix

7. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains a Target Address of a global address of TN1 that is not covered by an on-link prefix.

Observable Results:

- *Parts A-C*

  In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop. The HUT should respond to the second Echo Request directly on-link to TN1, indicating the HUT processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.3: Redirected On-link: Invalid (Hosts Only)

Purpose: Verify that a host properly processes invalid Redirect messages when redirected on-link.

References:

- [ND] – Sections 4.5 and 8.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host MUST silently discard any received Redirect message that does not satisfy all of the following validity checks:

- IP Source Address is a link-local address. Routers must use their link-local address as the source for Router Advertisement and Redirect messages so that hosts can uniquely identify routers.
- The IP Hop limit field has a value of 255 (indicating the packet was sent from a router on the same physical network).
- If the message includes an IP Authentication Header, the message authenticates correctly.
- ICMPv6 Checksum is valid.
- ICMPv6 Code is 0.
- ICMPv6 length (derived from the IP length) is 40 or more octets.
- The IP Source Address of the Redirect message is the same as the current first-hop router for the specified ICMPv6 Destination Address.
- The ICMPv6 Destination Address field in the Redirect message does not contain a multicast address.
- The ICMPv6 Target Address is either a link-local address (when redirected to a router) or the same as the ICMPv6 Destination Address (when redirected to the on-link destination).
- All included options have a length that is greater than zero.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Redirect Source Address is Global

1. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TN1. The Redirect message contains an incorrect IPv6 Source Address (the off-link global address of TN2).
4. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.

5. Observe the packets transmitted by the HUT.

Part B: Redirect Source Address is not the current first-hop router

6. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an incorrect IPv6 Source Address (the link-local address of TR2).

Part C: Hop Limit is not 255

7. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an incorrect IPv6 Hop Limit of 254.

Part D: ICMPv6 Code is not 0

8. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an incorrect ICMPv6 Code of 1.

Part E: ICMPv6 Checksum is invalid

9. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an incorrect ICMPv6 Checksum.

Part F: ICMPv6 Destination Address is Multicast

10. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an ICMPv6 Destination Address of the All-nodes multicast address.

Part G: Target Address is Multicast

11. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains a Target Address of the All-nodes multicast address.

Part H: ICMPv6 length is less than 40 Octets

12. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an invalid ICMPv6 Length of 39 bytes.

Part I: Option has Length Zero

13. Repeat Steps 1 through 5 from Part A. In Step 3, the Redirect message contains an Option with length 0.

Observable Results:

- *Parts A-I*

  In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop, as it is the only router in the HUT's Default Router List. The HUT should also respond to the second Echo Request using TR1 as a first hop, indicating the HUT did not process the invalid Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.4: Redirected to Alternate Router: Valid (Hosts Only)

Purpose: Verify that a host properly processes valid Redirect messages when redirected to alternate router.

References:

- [ND] – Sections 4.6.1, 4.6.3, and 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for the destination, an implementation SHOULD create such an entry. The Redirected Header option is used in Redirect messages and contains all or part of the packet that is being redirected. The Target Link-layer Address option is used in Redirect and Neighbor Advertisement messages and contains the link-layer address for the target.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

IPv6 Destination Address	TLLA Option	Redirected Packet Option	Part
Link-local (HUT)	No	No	A
Link-local (HUT)	No	Yes	B
Link-local (HUT)	Yes	No	C
Link-local (HUT)	Yes	Yes	D
Global (HUT)	No	No	E
Global (HUT)	No	Yes	F
Global (HUT)	Yes	No	G
Global (HUT)	Yes	Yes	H
Global (HUT)	Yes	Yes	I

Procedure:

Parts A through H: Destination Addresses, TLLA Options, and Redirected Packet Options

1. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
4. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message

contains a Target Link-layer Address option or Redirected Packet option according to the table above.

5. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
6. Observe the packets transmitted by the HUT.
7. Repeat Steps 1 through 6 for each Part B through H, using the Redirect message detailed in the table above in Step 4.

Part I: Redirected to Router not in Default Router List

8. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
9. Observe the packets transmitted by the HUT.
10. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
11. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
12. Observe the packets transmitted by the HUT.

Observable Results:

- *Parts A-I*

  In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop. The HUT should respond to the second Echo Request using TR2 as a first hop, indicating the HUT processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.5: Redirected to Alternate Router: Suspicious (Hosts only)

Purpose: Verify that a host properly processes suspicious Redirect messages when redirected on-link.

References:

- [ND] – Sections 4.5, 4.6.3, 8.1, and 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for he destination, an implementation SHOULD create such an entry. Hosts can be redirected to a better first-hop but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMPv6 Target Address equal to the ICMPv6 Destination Address. If the ICMPv6 Target and Destination Addresses are both the same, the host MUST treat the Target as on-link. The Redirected Header option is used in Redirect messages and contains all or part of the packet that is being redirected.

The contents of the Reserved field, and any unrecognized options MUST be ignored.

A host MUST NOT consider a redirect invalid just because the Target Address of the redirect is not covered under one of the link's prefixes. Part of the semantics of the Redirect message is that the Target Address is on-link.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Option Unrecognized

1. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
4. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option. The Redirect message also contains an unrecognized option.
5. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
6. Observe the packets transmitted by the HUT.

Part B: Reserved Field is Non-zero

7. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
8. Observe the packets transmitted by the HUT.
9. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
10. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option. The Redirect message also contains a non-zero Reserved field.
11. TR1 forwards an Echo Request to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
12. Observe the packets transmitted by the HUT.

Observable Results:

- *Parts A-B*
  In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop. The HUT should respond to the second Echo Request using TR2 as a first hop, indicating the HUT processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.6: Redirected to Alternate Router: Invalid (Hosts Only)

Purpose: Verify that a host properly processes invalid Redirect messages when redirected on-link.

References:

- [ND] – Sections 4.5 and 8.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host MUST silently discard any received Redirect message that does not satisfy all of the following validity checks:

- IP Source Address is a link-local address. Routers must use their link-local address as the source for Router Advertisement and Redirect messages so that hosts can uniquely identify routers.
- The IP Hop limit field has a value of 255 (indicating the packet was sent from a router on the same physical network).
- If the message includes an IP Authentication Header, the message authenticates correctly.
- ICMPv6 Checksum is valid.
- ICMPv6 Code is 0.
- ICMPv6 length (derived from the IP length) is 40 or more octets.
- The IP Source Address of the Redirect message is the same as the current first-hop router for the specified ICMPv6 Destination Address.
- The ICMPv6 Destination Address field in the Redirect message does not contain a multicast address.
- The ICMPv6 Target Address is either a link-local address (when redirected to a router) or the same as the ICMPv6 Destination Address (when redirected to the on-link destination).
- All included options have a length that is greater than zero.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Redirect Source Address is Global

1. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
4. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains an incorrect IPv6 Source Address (the off-link global address of TN2).

5. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
6. Observe the packets transmitted by the HUT.

Part B: Redirect Source Address is not the current first-hop router

7. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an incorrect IPv6 Source Address (the link-local address of TR2).

Part C: Hop Limit is not 255

8. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an incorrect IPv6 Hop Limit of 254.

Part D: ICMPv6 Code is not 0

9. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an incorrect ICMPv6 Code of 1.

Part E: ICMPv6 Checksum is invalid

10. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an incorrect ICMPv6 Checksum.

Part F: ICMPv6 Destination Address is Multicast

11. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an ICMPv6 Destination Address of the All-nodes multicast address.

Part G: Target Address is Multicast

12. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains a Target Address of the All-nodes multicast address.

Part H: ICMPv6 length is less than 40 Octets

13. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an invalid IPv6 Length of 39 bytes.

Part I: Option has Length Zero

14. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains an Option with length 0.

Observable Results:

- *Parts A-I*

  In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop. The HUT should also respond to the second Echo Request using TR1 as a first hop, indicating the HUT did not process the invalid Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.7: Redirected Twice (Hosts Only)

Purpose: Verify that a host properly processes valid Redirect messages twice for the same destination.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for the destination, an implementation SHOULD create such an entry.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

1. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
4. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2.
5. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
6. Observe the packets transmitted by the HUT.
7. TR3 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
8. TR2 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR3.
9. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
10. Observe the packets transmitted by the HUT.

Observable Results:

Step 2: The HUT should respond to the Echo Request using TR1 as a first hop, as it is the only router in the HUT's Default Router List.
Step 6: The HUT should respond to the Echo Request using TR2 as a first hop, indicating the HUT processed the Redirect message.

Step 10: The HUT should respond to the Echo Request using TR3 as a first hop, indicating the HUT processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.8: Invalid Option (Hosts Only)

Purpose: Verify that a host ignores invalid options in Redirect messages and processes the remainder of the Redirect normally.

References:

- [ND] – Section 8.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The contents of any defined options that are not specified to be used with Redirect messages MUST be ignored and the packet processed as normal. The only defined options that may appear are the Target Link-Layer Address option and the Redirected Header option.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Path MTU Option

1. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a Router Advertisement with a non-zero Router Lifetime and a Source Link-layer Address option.
4. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains a Path MTU option.
5. TR1 forwards an Echo Request from TN1 to the HUT. The Source Address is the off-link global address of TN1. The Destination Address is the global address of the HUT.
6. Observe the packets transmitted by the HUT.

Part B: Prefix Information Option

7. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains a Prefix Information option.

Part C: Source Link-layer Address Option

8. Repeat Steps 1 through 6 from Part A. In Step 4, the Redirect message contains a Source Link-layer Address option.

Observable Results:

- *Parts A-C*

In all Parts, the HUT should respond to the first Echo Request using TR1 as a first hop. The HUT should respond to the second Echo Request using TR2 as a first hop, indicating the HUT ignored the invalid option and processed the Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.9: No Destination Cache Entry (Hosts Only)

Purpose: Verify that a host properly processes a Redirect message when there is no entry for the destination in the host's Destination Cache.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host receiving a valid redirect SHOULD update its Destination Cache accordingly so that subsequent traffic goes to the specified target. If no Destination Cache entry exists for the destination, an implementation SHOULD create such an entry.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

Procedure:

1. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-Layer option with the link-layer address of TR2.
2. TR1 forwards an Echo Request from TN1 to the HUT. The IPv6 Source Address is the off-link global address of TN1. The IPv6 Destination Address is the global address of the HUT.
3. Observe the packets transmitted by the HUT.

Observable Results:

Step 3: The HUT should respond to the Echo Request using TR2 as the first-hop, indicating the HUT processed the Redirect message and created a Destination Cache entry.

Possible Problems:

- None.

Test v6LC.2.3.10: Neighbor Cache Updated, No Neighbor Cache Entry (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache entry upon receipt of a valid ICMP Redirect Message.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If a host receives a Redirect message containing a Target Link-Layer Address option the host either creates or updates the Neighbor Cache entry for the target. In both cases the cached link-layer address is copied from the Target Link-Layer option. If a Neighbor Cache entry is created for the target its reachability MUST be set to STALE as specified in section 7.3.3. If a cache entry already existed and it is updated with a different link-layer address, its reachability state MUST also be set to STALE. If the link-layer address is the same as that already in the cache, the cache entry's state remains unchanged.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

TLLA Option	Redirected Packet Option	New NC State	Link-layer Address	Part
No	No	No NCE	Unchanged	A
Yes	No	STALE	Updated	B
Yes	Yes	STALE	Updated	C
Yes	Yes, packet > 1280	STALE	Updated	D

Procedure:

Part A: No TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
2. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
3. TR2 transmits a link-local Echo Request to the HUT.
4. Wait 2 seconds and observe the packets transmitted by the HUT.

Part B: TLLA Option, No Redirected Packet Option, Link-layer Address Updated

5. Repeat Steps 1 through 4 from Part A, using the Redirect message detailed in the table above.

Part C: TLLA Option, Redirected Packet Option, Link-layer Address Updated

6. Repeat Steps 1 through 4 from Part A, using the Redirect message detailed in the table above.

Part D: TLLA Option, Oversized Redirected Packet Option, Link-layer Address Updated

7. Repeat Steps 1 through 4 from Part A, using the Redirect message detailed in the table above.

Observable Results:

- *Part A*

  Step 4: The HUT should wait to send a multicast Neighbor Solicitation for TR2 until it receives the Echo Request in Step 3, indicating the HUT had no NCE for TR2 before Step 3.
- *Parts B through D*

  Step 4: Because the HUT's NCE for TR2 is in state **STALE**, the HUT should send an Echo Reply to TR2 using the new Link-Layer address and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the HUT should send a unicast Neighbor Solicitation to TR2.

Possible Problems:

- None.

Test v6LC.2.3.11: Neighbor Cache Updated from State INCOMPLETE (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache entry upon receipt of a valid ICMP Redirect Message.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If a host receives a Redirect message containing a Target Link-Layer Address option the host either creates or updates the Neighbor Cache entry for the target. In both cases the cached link-layer address is copied from the Target Link-Layer option. If a Neighbor Cache entry is created for the target its reachability MUST be set to STALE as specified in section 7.3.3. If a cache entry already existed and it is updated with a different link-layer address, its reachability state MUST also be set to STALE. If the link-layer address is the same as that already in the cache, the cache entry's state remains unchanged.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

TLLA Option	Redirected Packet Option	New NC State	Link-layer Address	Part
No	No	INCOMPLETE	Unchanged	A
Yes	No	STALE	Updated	B
Yes	Yes	STALE	Updated	C
Yes	Yes, packet > 1280	STALE	Updated	D

Procedure:

Part A: No TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. TR2 transmits a link-local Echo Request to the HUT. TR2 does not reply to Neighbor Solicitations.
2. Observe packets transmitted by the HUT.
3. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
4. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
5. Observe the packets transmitted by the HUT.

Part B: TLLA Option, No Redirected Packet Option, Link-layer Address Updated

6. TR2 transmits a link-local Echo Request to the HUT. TR2 does not reply to Neighbor Solicitations.
7. Observe the packets transmitted by the HUT.
8. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
9. Observe the packets transmitted by the HUT.

Part C: TLLA Option, Redirected Packet Option, Link-layer Address Updated

10. Repeat Steps 6 through 9 from Part B, using the Redirect message detailed in the table above.

Part D: TLLA Option, Oversized Redirected Packet Option, Link-layer Address Updated

11. Repeat Steps 6 through 9 from Part B, using the Redirect message detailed in the table above.

Observable Results:

- *Part A*

  Step 2: The HUT should send a multicast Neighbor Solicitation for TR2, indicating the HUT has an NCE for TR2 in state **INCOMPLETE**.
  Step 5: The HUT should still send multicast Neighbor Solicitations for TR2, indicating the HUT still has an NCE for TR2 in state **INCOMPLETE**.
- *Parts B through D*

  Step 7: The HUT should send a multicast Neighbor Solicitation for TR2, indicating the HUT has an NCE for TR2 in state **INCOMPLETE**.
  Step 9: Because the HUT's NCE for TR2 is in state **STALE**, the HUT should send an Echo Reply to TR2 using the new Link-Layer address and enter state **DELAY**. After DELAY_FIRST_PROBE_TIME, the HUT should send a unicast Neighbor Solicitation to TR2.

Possible Problems:

- None.

Test v6LC.2.3.12: Neighbor Cache Updated from State REACHABLE (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache entry upon receipt of a valid ICMP Redirect Message.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If a host receives a Redirect message containing a Target Link-Layer Address option the host either creates or updates the Neighbor Cache entry for the target. In both cases the cached link-layer address is copied from the Target Link-Layer option. If a Neighbor Cache entry is created for the target its reachability MUST be set to STALE as specified in section 7.3.3. If a cache entry already existed and it is updated with a different link-layer address, its reachability state MUST also be set to STALE. If the link-layer address is the same as that already in the cache, the cache entry's state remains unchanged.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

TLLA Option	Redirected Packet Option	New NC State	Link-layer Address	Part
No	No	REACHABLE	Unchanged	A
Same	No	REACHABLE	Unchanged	B
Different	No	STALE	Updated	C
Different	Yes	STALE	Updated	D
Different	Yes, packet > 1280	STALE	Updated	E

Procedure:

Part A: No TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. TR2 transmits a link-local Echo Request to the HUT.
2. Observe the packets transmitted by the HUT.
3. TR2 transmits a solicited Neighbor Advertisement in response to any Neighbor Solicitations from the HUT.
4. Observe the packets transmitted by the HUT.
5. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
6. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)

7. TR2 transmits a link-local Echo Request to the HUT.
8. Observe the packets transmitted by the HUT.
9. Wait 2 seconds.
10. Wait DELAY_FIRST_PROBE_TIME. (5 seconds)
11. Observe the packets transmitted by the HUT.

Part B: TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. Repeat Steps 1 through 11 from Part A, using the Redirect message detailed in the table above.

Part C: TLLA Option, No Redirected Packet Option, Link-layer Address Updated

2. Repeat Steps 1 through 11 from Part A, using the Redirect message detailed in the table above.

Part D: TLLA Option, Redirected Packet Option, Link-layer Address Updated

3. Repeat Steps 1 through 11 from Part A, using the Redirect message detailed in the table above.

Part E: TLLA Option, Oversized Redirected Packet Option, Link-layer Address Updated

4. Repeat Steps 1 through 11 from Part A, using the Redirect message detailed in the table above.

Observable Results:

- *Parts A and B*
  Step 2: The HUT should create a Neighbor Cache Entry for TR2 and set the state of the Entry to INCOMPLETE. The HUT should transmit multicast Neighbor Solicitations to TR2.
  Step 4: After receiving the solicited Neighbor Advertisement from TR2, the HUT should update its Neighbor Cache Entry for TR2 to REACHABLE and transmit an Echo Reply. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR2.
  Step 8: The HUT should respond with an Echo Reply.
  Step 11: The HUT should not send any Neighbor Solicitations, indicating the HUT had a NCE for TR2 in state REACHABLE.
- *Parts C through E*
  Step 2: The HUT should create a Neighbor Cache Entry for TR2 and set the state of the Entry to INCOMPLETE. The HUT should transmit multicast Neighbor Solicitations to TR2.
  Step 4: After receiving the solicited Neighbor Advertisement from TR2, the HUT should update its Neighbor Cache Entry for TR2 to REACHABLE and transmit an Echo Reply. After DELAY_FIRST_PROBE_TIME, the HUT should not send a unicast Neighbor Solicitation to TR2.
  Step 8: The HUT should respond with an Echo Reply sent to the updated link-layer address.
  Step 11: The HUT should send a unicast Neighbor Solicitation for TR2, indicating the HUT had a NCE for TR2 in state STALE.

Possible Problems:

- None.

Test v6LC.2.3.13: Neighbor Cache Updated from State STALE (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache entry upon receipt of a valid ICMP Redirect Message.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If a host receives a Redirect message containing a Target Link-Layer Address option the host either creates or updates the Neighbor Cache entry for the target. In both cases the cached link-layer address is copied from the Target Link-Layer option. If a Neighbor Cache entry is created for the target its reachability MUST be set to STALE as specified in section 7.3.3. If a cache entry already existed and it is updated with a different link-layer address, its reachability state MUST also be set to STALE. If the link-layer address is the same as that already in the cache, the cache entry's state remains unchanged.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

TLLA Option	Redirected Packet Option	New NC State	Link-layer Address	Part
No	No	STALE	Unchanged	A
Same	No	STALE	Unchanged	B
Different	No	STALE	Updated	C
Different	Yes	STALE	Updated	D
Different	Yes, packet > 1280	STALE	Updated	E

Procedure:

Part A: No TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. TR2 transmits an unsolicited Router Advertisement with a Source Link-layer Address option to the all-nodes multicast address.
2. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
3. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
4. TR2 transmits a link-local Echo Request to the HUT.
5. Observe the packets transmitted by the HUT.
6. Wait 2 seconds.

7. Wait DELAY_FIRST_PROBE_TIME. (5 seconds)
8. Observe the packets transmitted by the HUT.

Part B: TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

9. Repeat Steps 1 through 8 from Part A, using the Redirect message detailed in the table above.

Part C: TLLA Option, No Redirected Packet Option, Link-layer Address Updated

10. Repeat Steps 1 through 8 from Part A, using the Redirect message detailed in the table above.

Part D: TLLA Option, Redirected Packet Option, Link-layer Address Updated

11. Repeat Steps 1 through 8 from Part A, using the Redirect message detailed in the table above.

Part E: TLLA Option, Oversized Redirected Packet Option, Link-layer Address Updated

12. Repeat Steps 1 through 8 from Part A, using the Redirect message detailed in the table above.

Observable Results:

- *Parts A and B*
  Step 5: The HUT should respond with an Echo Reply.
  Step 8: The HUT should send a unicast Neighbor Solicitation for TR2, indicating the HUT had a NCE for TR2 in state **STALE**.
- *Parts C through E*
  Step 5: The HUT should respond with an Echo Reply sent to the updated link-layer address.
  Step 8: The HUT should send a unicast Neighbor Solicitation for TR2, indicating the HUT had a NCE for TR2 in state **STALE**.

Possible Problems:

- None.

Test v6LC.2.3.14: Neighbor Cache Updated from State PROBE (Hosts Only)

Purpose: Verify that a host properly updates its Neighbor Cache entry upon receipt of a valid ICMP Redirect Message.

References:

- [ND] – Section 8.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: If a host receives a Redirect message containing a Target Link-Layer Address option the host either creates or updates the Neighbor Cache entry for the target. In both cases the cached link-layer address is copied from the Target Link-Layer option. If a Neighbor Cache entry is created for the target its reachability MUST be set to STALE as specified in section 7.3.3. If a cache entry already existed and it is updated with a different link-layer address, its reachability state MUST also be set to STALE. If the link-layer address is the same as that already in the cache, the cache entry's state remains unchanged.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part. The following table details the Redirect message transmitted in each Part:

TLLA Option	Redirected Packet Option	New NC State	Link-layer Address	Part
No	No	PROBE	Unchanged	A
Same	No	PROBE	Unchanged	B
Different	No	STALE	Updated	C
Different	Yes	STALE	Updated	D
Different	Yes, packet > 1280	STALE	Updated	E

Procedure:

Part A: No TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

1. TR2 transmits an unsolicited Router Advertisement with a Source Link-layer Address option to the all-nodes multicast address.
2. TR2 transmits an unsolicited Neighbor Advertisement for its link-local address to the HUT.
3. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR).
4. TR2 transmits an Echo Request from its link-local address to the HUT.
5. Wait DELAY_FIRST_PROBE_TIME. (5 seconds)
6. TR1 transmits a Redirect message to the NUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.

7. Observe the packets transmitted by the HUT.

Part B: TLLA Option, No Redirected Packet Option, Link-layer Address Unchanged

8. Repeat Steps 1 through 7 from A, using the Redirect message detailed in the table above.

Part C: TLLA Option, No Redirected Packet Option, Link-layer Address Updated

9. TR2 transmits an unsolicited Router Advertisement with a Source Link-layer Address option to the all-nodes multicast address.
10. TR2 transmits an unsolicited Neighbor Advertisement for its link-local address to the HUT.
11. Wait (REACHABLE_TIME * MAX_RANDOM_FACTOR).
12. TR2 transmits an Echo Request from its link-local address to the HUT.
13. Observe the packets transmitted by the HUT.
14. Wait DELAY_FIRST_PROBE_TIME. (5 seconds)
15. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2. The Redirect message contains a Target Link-layer Address option or Redirected Packet option according to the table above.
16. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
17. TR2 transmits a link-local Echo Request to the HUT.
18. Observe the packets transmitted by the HUT.
19. Wait 2 seconds.
20. Wait DELAY_FIRST_PROBE_TIME. (5 seconds)
21. Observe the packets transmitted by the HUT.

Part D: TLLA Option, Redirected Packet Option, Link-layer Address Updated

22. Repeat Steps 10 through 21 from C, using the Redirect message detailed in the table above.

Part E: TLLA Option, Oversized Redirected Packet Option, Link-layer Address Updated

23. Repeat Steps 10 through 21 from C, using the Redirect message detailed in the table above.

Observable Results:

- *Parts A and B*

  Step 7: The HUT should transmit a unicast Neighbor Solicitation for TR2, indicating the HUT had a NCE for TR2 in state **PROBE**.
- *Parts C through E*

  Step 13: The HUT should respond with an Echo Reply.
  Step 18: The HUT should respond with an Echo Reply sent to the updated link-layer address.
  Step 21: The HUT should send a unicast Neighbor Solicitation for TR2, indicating the HUT had a NCE for TR2 in state **STALE**.

Possible Problems:

- None.

Test v6LC.2.3.15: Invalid Redirect does not Update Neighbor Cache (Hosts Only)

Purpose: Verify that a host properly processes invalid Redirect messages when redirected on-link.

References:

- [ND] – Sections 8.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A host MUST silently discard any received Redirect message that does not satisfy all of the following validity checks:

- IP Source Address is a link-local address. Routers must use their link-local address as the source for Router Advertisement and Redirect messages so that hosts can uniquely identify routers.
- The IP Hop limit field has a value of 255 (indicating the packet was sent from a router on the same physical network).
- If the message includes an IP Authentication Header, the message authenticates correctly.
- ICMPv6 Checksum is valid.
- ICMPv6 Code is 0.
- ICMPv6 length (derived from the IP length) is 40 or more octets.
- The IP Source Address of the Redirect message is the same as the current first-hop router for the specified ICMPv6 Destination Address.
- The ICMPv6 Destination Address field in the Redirect message does not contain a multicast address.
- The ICMPv6 Target Address is either a link-local address (when redirected to a router) or the same as the ICMPv6 Destination Address (when redirected to the on-link destination).
- All included options have a length that is greater than zero.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. Wait at least 3 seconds (MAX_MULTICAST_SOLICIT * RETRANS_TIMER) after any previous cleanup to make sure all previous NCE's are in state No NCE. The Common Cleanup Procedure is performed after each part.

Procedure:

Part A: Redirect Source Address is Global

1. TR1 transmits a Redirect message to the HUT. The ICMPv6 Destination Address is the global address of TN1. The Target Address is the link-local address of TR2. The Redirect message contains an incorrect IPv6 Source Address (the off-link global address of TN2).
2. Wait (RETRANS_TIMER * MAX_*CAST_SOLICIT). (3 seconds)
3. TR2 transmits a link-local Echo Request to the HUT.
4. Wait 2 seconds.
5. Observe the packets transmitted by the HUT.

Part B: Redirect Source Address is not the current first-hop router

6. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an incorrect IPv6 Source Address (the link-local address of TR2).

Part C: Hop Limit is not 255

7. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an incorrect IPv6 Hop Limit of 254.

Part D: ICMPv6 Code is not 0

8. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an incorrect ICMPv6 Code of 1.

Part E: ICMPv6 Checksum is invalid

9. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an incorrect ICMPv6 Checksum.

Part F: ICMPv6 Destination Address is Multicast

10. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an ICMPv6 Destination Address of the all-nodes multicast address.

Part G: Target Address is Multicast

11. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains a Target Address of the All-nodes multicast address.

Part H: ICMPv6 length is less than 40 Octets

12. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an invalid IPv6 Length of 39 bytes.

Part I: Option has Length Zero

13. Repeat Steps 1 through 5 from Part A. In Step 1, the Redirect message contains an Option with length 0.

Observable Results:

- *Parts A through I*

  Step 5: The HUT should transmit a multicast Neighbor Solicitation for TR2, indicating the HUT did not create a NCE for TR2 upon reception of an invalid Redirect message.

Possible Problems:

- None.

Test v6LC.2.3.16: Redirect – Transmit (Routers Only)

Purpose: Verify that a router properly handles transmission of Redirect messages.

References:

- [ND] – Sections 8.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A router SHOULD send a redirect message, subject to rate limiting, whenever it forwards a packet that is not explicitly addressed to itself (i.e. a packet that is not source routed through the router) in which:

- the Source Address field of the packet identifies a neighbor, and

- the router determines that a better first-hop node resides on the same link as the sending node for the Destination Address of the packet being forwarded, and

- the Destination Address of the packet is not a multicast address, and

The transmitted redirect packet contains, consistent with the message format given in Section 4.5:

- In the Target Address field: the address to which subsequent packets for the destination SHOULD be sent. If the target is a router, that router's link-local address MUST be used. If the target is a host the target address field MUST be set to the same value as the Destination Address field.

- In the Destination Address field: the destination address of the invoking IP packet.

- In the options:

    - Target Link-Layer Address option: link-layer address of the target, if known.

    - Redirected Header: as much of the forwarded packet as can fit without the redirect packet exceeding 1280 octets in size.

The IPv6 Source address of a Redirect Message MUST be the link-local address assigned to the interface from which the message is sent.

Test Setup: Common Setup 1.1 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

i. TN2 is an on-link neighbor on Link B to TN1 (instead of residing on Link A depicted in Common Topology).
ii. RUT advertises prefix X on Link B.

Procedure:

Part A: Send Redirect

1. TN1 transmits an Echo Request to TN2's unicast global address with prefix X and a first hop through the RUT.
2. Observe the packets transmitted by the RUT.

Part B: Send Redirect to Alternate Router

3. Configure TN2 to be an off-link neighbor residing on Link A as depicted in the Common Topology.
4. TN1 transmits an Echo Request to TN2's unicast global address and a first hop through the RUT.
5. Observe the packets transmitted by the RUT.

Part C: Source not neighbor

6. TN1 transmits an Echo Request to TN2 with a first hop through the RUT. The Source Address is TN1's address with an off-link prefix.
7. Observe the packets transmitted by the RUT.

Part D: Destination Multicast

8. TN1 transmits an Echo Request to TN2's solicited-node multicast address with a first hop through the RUT.
9. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*

  Step 2: The RUT should transmit a Redirect message with the following values:

IPv6 Source	Link-Local address of RUT
IPv6 Destination	TN1's address (used in Echo Request's Source Address)
Target	TN2's unicast global address with prefix X
Destination	TN2's unicast global address with prefix X
TLL Option	TN2's link-layer address if known
Redirected Header	TN1's Echo Request without total packet exceeding 1280 bytes.

- *Part B*

  Step 5: The RUT should transmit a Redirect message with the following values:

IPv6 Source	Link-Local address of RUT
IPv6 Destination	TN1's address (used in Echo Request's Source Address)
Target	TR1's link-local address
Destination	TN2's unicast global address
TLL Option	TR1's link-layer address if known
Redirected Header	TN1's Echo Request without total packet exceeding 1280 bytes.

- *Parts C and D*

  Steps 7 and 9: The RUT should not send a Redirect message.

Possible Problems:

- The RUT may not support the generation of Redirect messages.

Test v6LC.2.3.17: Redirect – Receive (Routers Only)

Purpose: Verify that a router properly handles reception of Redirect messages.

References:

- [ND] – Sections 8.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A router MUST NOT update its routing tables upon receipt of a Redirect.

Test Setup: Common Setup 1.2 is performed at the beginning of each test part. The Common Cleanup Procedure is performed after each part.

i. Configure the RUT with a static route to TN2's Link B prefix through TR1.

Procedure:

1. TR1 forwards an Echo Request from TN2 to the RUT. The Destination Address is the global address of the RUT.
2. Observe the packets transmitted by the RUT.
3. TR1 transmits a Redirect message to the RUT. The ICMPv6 Destination Address is the global address of TN2. The Target Address is the link-local address of TR2.
4. TR1 forwards an Echo Request from TN2 to the RUT. The Destination Address is the global address of the RUT.
5. Observe the packets transmitted by the RUT.

Observable Results:

Step 2: The RUT should send an Echo Reply with a first hop through TR1.
Step 5: The RUT should still send an Echo Reply with a first hop through TR1, indicating the RUT did not change its routing table with information from TR1's Redirect message.

Possible Problems:

- None.

Section 3: RFC 2462

Scope

The following tests cover the IPv6 Stateless Address Autoconfiguration specification, Request For Comments 2462. These tests verify the process for generating a link-local address, the process for generating site-local and global addresses via stateless address autoconfiguration, and the Duplicate Address Detection procedure. The following tests also verify that a host correctly processes a Router Advertisement and correctly assigns lifetimes.

Overview

These tests are designed to verify the readiness of an IPv6 implementation vis-à-vis the IPv6 Stateless Address Autoconfiguration specification.

Default Packets

Echo Request

IPv6 Header Payload Length: 136 bytes Next Header: 58
ICMPv6 Header Type: 128 Code: 0

Router Advertisement

IPv6 Header Source Address: TR1's Link-Local Address Destination Address: All-Nodes multicast address Next Header: 58
ICMPv6 Header Type: 134 Code: 0 Hop Limit: 255 M Bit (managed): 0 O Bit (other): 0 Router Lifetime: 20 seconds Reachable Time: 10 seconds Retrans Timer: 1 second
Prefix Option Type: 3 L Bit (on-link flag): 1 A Bit (addr conf): 1 Valid Lifetime: 20 seconds Preferred Lifetime: 20 seconds

Group 1: Address Autoconfiguration and Duplicate Address Detection

Scope

The following tests cover Address autoconfiguration and duplicate address detection in IPv6.

Overview

The tests in this group verify conformance of the Address autoconfiguration and duplicate address detection with the IPv6 Stateless Address Autoconfiguration Specification.

Test v6LC.3.1.1: Address Autoconfiguration and Duplicate Address Detection

Purpose: Verify that a node can properly initialize on a network using address autoconfiguration and communicate with other on-link partners.

References:

- [ADDRCONF] – Sections 1, 5.3, 5.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: When a node initializes on a given link, it performs stateless address autoconfiguration and Duplicate Address Detection (DAD). To insure that all configured addresses are likely to be unique on a given link, hosts (routers are also expected to find their link-local addressing using this method) run the DAD algorithm on tentative addresses before assigning them to an interface. An address on which the Duplicate Address Detection procedure is applied is said to be tentative until the procedure has completed successfully. The Duplicate Address Detection algorithm is performed on all unicast addresses, independent of whether they are obtained via stateless or stateful autoconfiguration. DAD must not be performed on anycast addresses. In addition, routers perform DAD on all addresses prior to assigning them to an interface. The procedure for detecting duplicate addresses uses Neighbor Solicitation and Advertisement messages.

Test Setup: No Common Test Setup is performed. Default values include DupAddrDetectTransmits=1 and RetransTimer=1 second.

Procedure:

1. Initialize all the devices on Link B.
2. Allow time for all devices on Link B to perform stateless address autoconfiguration and DAD.
3. Transmit a DAD NS from TN1 with the Target Address set to the NUT’s link-local address.
4. Observe packet captures on Link B.

Observable Results:

Step 2: The NUT should perform DAD on its tentative address for its interface on Link B sending DupAddrDetectTransmits Neighbor Solicitations, every RetransTimer. The NUT should assign the tentative address to its interface.
Step 4: The NUT must transmit a DAD NA for its autoconfigured link-local address.

Possible Problems:

- None.

Test v6LC.3.1.2: Receiving DAD Neighbor Solicitations and Advertisements

Purpose: To verify that a node can properly process neighbor solicitations and advertisements performing Duplicate Address Detection while the node is also performing DAD.

References:

- [ADDRCONF] – Sections 1, 5.4, 5.4.1, 5.4.3 and 5.4.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: An interface on which Duplicate Address Detection is being performed must accept Neighbor Solicitations and Advertisement messages containing the tentative address in the Target Address field, but process such packets differently from those whose target address matches an address assigned to the interface. Other packets addressed to the tentative address should be silently discarded.

On receipt of a valid Neighbor Solicitation message on an interface, node behavior depends on whether the target address is tentative or not. If the source address of the Neighbor Solicitation is the unspecified address, the solicitation is from a node performing DAD. If the solicitation is from another node, the tentative address is a duplicate and should not be used (by either node).

On receipt of a valid Neighbor Advertisement message on an interface, node behavior depends on whether the target address is tentative. If the target address is tentative, the tentative address is not unique.

Test Setup: No Common Test Setup is performed. Default values include DupAddrDetectTransmits=1 and RetransTimer=1 second.

Neighbor Solicitation A

IPv6 Header Next Header: 58 Source Address: Unspecified Address Destination Address: Solicited multicast of the NUT's tentative Link-local Address Hop Limit: 255
Neighbor Solicitation Target Address: (See Below)

Neighbor Advertisement B

IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: all-nodes multicast address Hop Limit: 255
Neighbor Advertisement Router flag: 0 Solicited flag: 0 Override flag: 1 Target Address: (See Below) TLLOPT: TN1's MAC address

Procedure:

Part A: NUT receives DAD NS (target != NUT)

1. Initialize all devices on Link B.
2. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit DAD Neighbor Solicitation A with the Target Address set to TN1's link-local address.
3. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
4. Transmit a NS from TN1 with the Target Address set to the NUT's link-local address.
5. Observe packet captures on Link B.

Part B: NUT receives DAD NS (target == NUT)

6. Initialize all devices on Link B.
7. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit DAD Neighbor Solicitation A with the Target Address set to the NUT's tentative link-local address.
8. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
9. Transmit a NS from TN1 with the same target address used for the DAD NS in Step 7.
10. Observe packet captures on Link B.

Part C: NUT receives DAD NA (target != NUT)

11. Initialize all devices on Link B.
12. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit DAD Neighbor Advertisement B with a Target Address set to TN1's link-local address.
13. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
14. Transmit a NS from TN1 with the target address set to the NUT's link-local address.
15. Observe packet captures on Link B.

Part D: NUT receives DAD NA (target == NUT)

16. Initialize the devices on Link B.

17. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit DAD Neighbor Advertisement B with a Target Address set to the NUT's tentative link-local address and no TLL Option.
18. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
19. Transmit a NS from TN1 with the same target address used for the DAD NS in Step 17.
20. Observe packet captures on Link B.

Observable Results:

- *Part A*

  Step 3: The NUT should silently ignore the DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 5: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part B*

  Step 8: The NUT should receive more DAD NS messages than expected with its tentative link-local address as the Target address. The NUT should determine its tentative address is a duplicate and should not assign the tentative address to its interface.
  Step 10: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.
- *Part C*

  Step 13: The NUT should silently ignore DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 15: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part D*

  Step 18: The NUT should determine its tentative address is not unique and should not assign the tentative address to its interface.
  Step 20: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.

Possible Problems:

- None.

Test v6LC.3.1.3: Validation of DAD Neighbor Solicitations

Purpose: Verify that a node can properly ignore invalid neighbor solicitations while performing Duplicate Address Detection.

References:

- [ADDRCONF] – Section 5.4.1
- [ND] – Section 6.1.1

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node must silently discard any Neighbor Solicitation or Advertisement message that does not pass the validity checks specified in [ND].

A node MUST silently discard any received Neighbor Solicitation messages that do not satisfy all of the following validity checks:

- Hop Limit has a value of 255
- If Message includes an Authentication Header, it is authenticated correctly
- ICMP Checksum is valid
- ICMP code = 0
- ICMP length is 24 or more octets
- Target Address is not a multicast address
- If the IP Source Address is the unspecified address, the IP Destination Address is a solicited-node multicast address.
- If the IP Source Address is the unspecified address, there is no source link-layer address option in the message.
- The contents of the Reserved field, and of any unrecognized options, MUST be ignored.

Test Setup: No Common Test Setup is performed. Default values include DupAddrDetectTransmits=1 and RetransTimer=1 second.

Neighbor Solicitation A

IPv6 Header Next Header: 58 Source Address: Unspecified Address Destination Address: Solicited multicast of the NUT’s tentative Link-local Address Hop Limit: 255
Neighbor Solicitation

Target Address: NUT's tentative link-local address

Procedure:

Part A: NUT receives invalid DAD NS (ICMP length < 24 octets)

1. Initialize all devices on Link B.
2. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the ICMP length set to 16.
3. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
4. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
5. Observe packet captures on Link B.

Part B: NUT receives invalid DAD NS (HopLimit !=255)

6. Initialize all devices on Link B.
7. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the Hoplimit set to 254.
8. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
9. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
10. Observe packet captures on Link B.

Part C: NUT receives invalid DAD NS (Dst = NUT's tentative address)

11. Initialize all devices on Link B.
12. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the Destination address set to the NUT's tentative link-local address.
13. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
14. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
15. Observe packet captures on Link B.

Part D: NUT receives invalid DAD NS (Dst = allnode)

16. Initialize all devices on Link B.
17. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with Destination address set to the all-nodes multicast address.
18. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
19. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
20. Observe packet captures on Link B.

Part E: NUT receives invalid DAD NS (ICMP code!= zero)

21. Initialize all devices on Link B.
22. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the ICMP code set to 1.
23. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
24. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
25. Observe packet captures on Link B.

Part F: NUT receives invalid DAD NS (Invalid Checksum)

26. Initialize all devices on Link B.

27. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with an invalid ICMP Checksum.
28. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
29. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
30. Observe packet captures on Link B.

Part G: NUT receives invalid DAD NS (target == multicast address)

31. Initialize all devices on Link B.
32. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the Target Address set to the solicited multicast of the NUT's tentative link-local address.
33. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
34. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
35. Observe packet captures on Link B.

Part H: NUT receives invalid DAD NS (contains SLL)

36. Initialize all devices on Link B.
37. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A containing a SLL Option set to TN1's MAC address.
38. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
39. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
40. Observe packet captures on Link B.

Part I: NUT receives valid DAD NS (Reserved Field)

41. Initialize all devices on Link B.
42. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A with the Reserved field set to 0xFFFFFFFF.
43. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
44. Transmit a Solicited NS from TN1 with the same target address used for the DAD Neighbor Solicitation A.
45. Observe packet captures on Link B.

Part J: NUT receives valid DAD NS (contains TLL)

46. Initialize the devices on Link B.
47. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Solicitation A containing a TLL Option set to TN1's MAC address.
48. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
49. Transmit a Solicited NS from TN1 with the same target address used for the DAD Neighbor Solicitation A.
50. Observe packet captures on Link B.

Observable Results:

- *Part A*
  Step 3: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 5: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part B*
  Step 8: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 10: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part C*
  Step 13: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 15: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part D*
  Step 18: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 20: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part E*
  Step 23: The NUT should silently ignore the DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 25: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part F*
  Step 28: The NUT should silently ignore the DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 30: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part G*
  Step 33: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 35: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part H*
  Step 38: The NUT should silently ignore the invalid DAD NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 40: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part I*
  Step 43: The NUT should ignore the contents of the Reserved field. The NUT should not assign the tentative address to its interface.
  Step 45: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.
- *Part J*
  Step 43: The NUT should ignore any options they do not recognize and continue processing the message. The NUT should not assign the tentative address to its interface.
  Step 45: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.

Possible Problems:

- None.

Test v6LC.3.1.4: Receiving Invalid Neighbor Advertisements

Purpose: Verify that a node can properly ignore invalid neighbor advertisements while performing Duplicate Address Detection.

References:

- [ADDRCONF] – Section 5.4.1
- [ND] – Section 7.1.2

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: A node must silently discard any Neighbor Solicitation or Advertisement message that does not pass the validity checks specified in [ND].

A node MUST silently discard any received Neighbor Advertisement messages that do not satisfy all of the following validity checks:

- IP Hop Limit has a value of 255
- If Message includes an Authentication Header, it is authenticated correctly
- ICMP Checksum is valid
- ICMP code = 0
- ICMP length is 24 or more octets
- Target Address is not a multicast address
- If the IP Destination Address is multicast, the Solicited Flag = 0
- All included options have a length > 0

The contents of the Reserved field, and of any unrecognized options, MUST be ignored.

Test Setup: No Common Test Setup is performed. Default values include DupAddrDetectTransmits=1 and RetransTimer=1 second.

Neighbor Advertisement A
IPv6 Header Next Header: 58 Source Address: TN1's Link-local Address Destination Address: all-nodes multicast address Hop Limit: 255
Neighbor Advertisement Router flag: 0 Solicited flag: 0

Override flag: 1 Target Address: NUT's tentative link-local address TLLOPT: TN1's MAC address

Procedure:

Part A: NUT receives invalid DAD NA (ICMP length < 24 octets)

1. Initialize all devices on Link B.
2. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the ICMP length set to 16.
3. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
4. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
5. Observe packet captures on Link B.

Part B: NUT receives invalid DAD NA (HopLimit != 255)

6. Initialize all devices on Link B.
7. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the Hoplimit set to 254.
8. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
9. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
10. Observe packet captures on Link B.

Part C: NUT receives invalid DAD NA (ICMP code!= zero)

11. Initialize all devices on Link B.
12. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the ICMP code set to 1.
13. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
14. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
15. Observe packet captures on Link B.

Part D: NUT receives invalid DAD NA (Invalid Checksum)

16. Initialize all devices on Link B.
17. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with an invalid ICMP Checksum.
18. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
19. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
20. Observe packet captures on Link B.

Part E: NUT receives invalid DAD NA (SolicitedFlag ==1)

21. Initialize all devices on Link B.
22. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the Solicited flag set to 1.
23. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.

24. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
25. Observe packet captures on Link B.

Part F: NUT receives invalid DAD NA (target == multicast address)

26. Initialize all devices on Link B.
27. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the Target Address set to the solicited multicast of the NUT's tentative link-local address.
28. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
29. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
30. Observe packet captures on Link B.

Part G: NUT receives invalid DAD NA (option length ==zero)

31. Initialize all devices on Link B.
32. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the TLLOPT Length set to 0.
33. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
34. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
35. Observe packet captures on Link B.

Part H: NUT receives valid DAD NA (Reserved Field)

36. Initialize all devices on Link B.
37. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A with the Reserved field set to 0x1FFFFFFF.
38. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
39. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
40. Observe packet captures on Link B.

Part I: NUT receives valid DAD NA (contains SLL)

41. Initialize the devices on Link B.
42. After TN1 receives a DAD NS message from the NUT, configure TN1 to transmit Neighbor Advertisement A containing a SLL Option set to TN1's MAC address.
43. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
44. Transmit valid Solicited NS from TN1 with the target address set to the NUT's link-local address.
45. Observe packet captures on Link B.

Observable Results:

- *Part A*

  Step 3: The NUT should silently ignore the invalid DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 5: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part B*

  Step 8: The NUT should silently ignore the invalid DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 10: The NUT must transmit a Solicited NA for its autoconfigured link-local address.

- *Part C*
  Step 13: The NUT should silently ignore the invalid DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 15: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part D*
  Step 18: The NUT should silently ignore the invalid DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 20: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part E*
  Step 23: The NUT should silently ignore the DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 25: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part F*
  Step 28: The NUT should silently ignore the DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 30: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part G*
  Step 33: The NUT should silently ignore the invalid DAD NA. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 35: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part H*
  Step 38: The NUT should ignore the contents of the Reserved field. The NUT should not assign the tentative address to its interface.
  Step 40: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.
- *Part I*
  Step 43: The NUT should ignore any options they do not recognize and continue processing the message. The NUT should not assign the tentative address to its interface.
  Step 45: The NUT must NOT transmit a Solicited NA for its autoconfigured link-local address.

Possible Problems:

- None.

Test v6LC.3.1.5: Receiving Neighbor Solicitations for Address Resolution

Purpose: Verify that a node can properly ignore neighbor solicitations performing address resolution while performing Duplicate Address Detection.

References:

- [ADDRCONF] – Sections 1, 5.4.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: On receipt of a valid Neighbor Solicitation message on an interface, node behavior depends on whether the target address is tentative or not. If the target address is tentative and the solicitation's sender is performing address resolution on the target, the solicitation should be silently ignored.

Test Setup: No Common Test Setup is performed. Default values include DupAddrDetectTransmits=1 and RetransTimer=1 second.

Neighbor Solicitation A

IPv6 Header Next Header: 58 Source Address: TN1's link-local address Destination Address: Solicited multicast of the NUT's tentative Link-local Address Hop Limit: 255
Neighbor Solicitation Target Address: NUT's tentative link-local address

Procedure:

Part A: NUT receives NS (src == unicast)

1. Initialize all devices on Link B.
2. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit Neighbor Solicitation A.
3. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate

Address Detection.
4. Transmit valid NS from TN1 with the target address set to the NUT’s link-local address.
5. Observe packet captures on Link B.

Part B: NUT receives NS (Src == unicast && Dst == NUT’s tentative address)

6. Initialize all devices on Link B.
7. After TN1 receives a DAD NS message from the NUT. Configure TN1 to transmit Neighbor Solicitation A with the Destination Address set to the NUT’s tentative link-local address.
8. Allow time for all devices on Link B to perform stateless address autoconfiguration and Duplicate Address Detection.
9. Transmit valid NS from TN1 with the target address set to the NUT’s link-local address.
10. Observe packet captures on Link B.

Observable Results:

- *Part A*
  Step 3: The NUT should silently ignore the NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 5: The NUT must transmit a Solicited NA for its autoconfigured link-local address.
- *Part B*
  Step 8: The NUT should silently ignore the NS. The NUT should complete the DAD process and assign the tentative address to its interface.
  Step 10: The NUT must transmit a Solicited NA for its autoconfigured link-local address.

Possible Problems:

- None.

Group 2: Router Advertisement Processing and Address Lifetime

Scope

The following tests cover Router Advertisement processing and address lifetime expiry in IPv6.

Overview

The tests in this group verify conformance creating global addresses, processing Router Advertisements and expiring an address with the IPv6 Stateless Address Autoconfiguration Specification.

Test v6LC.3.2.1: Address Lifetime Expiry (Hosts Only)

Purpose: Verify that a host can properly handle expired or invalid addresses.

References:

- [ADDRCONF] – Sections 4.1 and 5.5.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: IPv6 addresses are leased to an interface for a fixed (possibly infinite) length of time. An address has a lifetime that indicates how long that address is bound to that interface. When a valid lifetime of an IPv6 address expires it becomes invalid. Initially, an address is "preferred", meaning that its use in arbitrary communication is unrestricted. Later, an address becomes "deprecated" in anticipation that its current interface binding will become invalid. An invalid address MUST NOT be used as a source address in outgoing communications and MUST NOT be recognized as a destination on a receiving interface.

Test Setup: No Common Test Setup is performed.

Procedure:

1. Initialize all devices on Link B.
2. Configure TR1 to send out ONE Router Advertisement on Link B with Prefix "X" with a valid lifetime set to 40 seconds.
3. Allow time for the HUT to perform stateless address autoconfiguration and Duplicate Address Detection.
4. Configure TR1 to transmit a NS message for address resolution with the target address set to the NUT's global address for Prefix "X". Observe packets on Link B.
5. Wait 35 seconds.
6. Repeat Step 4.
7. Wait 10 seconds.
8. Configure TR1 to transmit a NS message for address resolution with the target address set to the NUT's global address for Prefix "X". Observe packets on Link B.

Observable Results:

Step 4: The HUT must transmit a Solicited NA for its autoconfigured global address.
Step 6: The HUT must transmit a Solicited NA for its autoconfigured global address.
Step 8: The HUT must NOT transmit a Solicited NA for its autoconfigured global address using Prefix "X".

Possible Problems:

- None.

Test v6LC.3.2.2: Multiple Prefixes and Network Renumbering (Hosts only)

Purpose: To verify that a host configured with multiple prefixes can communicate with another host on a different network when its site has been renumbered.

References:

- [ADDRCONF] – Section 4.1
- [ND] – Section 6.3.4, 6.3.5, 12

Resource Requirements:

- Packet generator
- Monitor to capture packets
- Ping6 implementations

Discussion: Multiple prefixes can be associated with the same link. By default, hosts learn all on-link prefixes from Router Advertisements. A host with multiple on-link prefixes should be able to communicate using any configured prefix, as long as its lifetime is still valid. Further, the Neighbor Discovery protocol together with IPv6 Address Autoconfiguration provides mechanisms to aid in renumbering – new prefixes and addresses can be introduced and old ones can be deprecated and removed. Address leasing facilitates site renumbering by providing a mechanism to time-out address assigned to interfaces in hosts.

Test Setup: Initialize all devices on Link B. Perform common Test Setup 1.1, the prefix lifetime will be configured to 20 seconds.

Procedure:

1. Configure TR1 to discontinue to send RA's for Prefix "X".
2. Configure TR1 to send out Router Advertisements on Link B with Prefix "Y" with a Valid Lifetime of 30 seconds.
3. Wait 10 seconds allowing time for the HUT to configure a new global address with the new prefix and for Duplicate Address Detection to be performed.
4. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X". Observe packets on Link B.
5. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "Y". Observe packets on Link B.
6. Wait 11 seconds allowing enough time to elapse so that Prefix "X" has timed out and Prefix "Y" has not timed out.
7. Repeat Step 4.
8. Repeat Step 5.
9. Configure TR1 to discontinue sending RA's for Prefix "Y". Wait 10 seconds allowing enough time to elapse so the Prefix "Y" has timed out.
10. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "Y". Observe packets on Link B.

Observable Results:

Step 2: The HUT should configure a new global address with the new prefix, Prefix “Y”.
Step 4: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
Step 5: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix “Y”.
Step 7: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
Step 8: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix “Y”.
Step 10: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “Y”.

Possible Problems:

None.

Test v6LC.3.2.3: Prefix-Information Option Processing (Hosts Only)

Purpose: Verify that a host properly processes the Prefix Information Option in the Router Advertisement.

References:

- [ADDRCONF] – Section 5.5.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: Router Advertisements contain zero or more Prefix information options that contain information used by stateless address autoconfiguration to generate site-local and global addresses. One Prefix Information option field, the "autonomous address-configuration flag", indicates whether or not the option even applies to stateless autoconfiguration. If it does, additional option fields contain a subnet prefix together with lifetime values indicating how long addresses created from the prefix remain preferred and valid.

For each Prefix-Information option in the Router Advertisement: Silently ignore the Prefix Information Option if the Autonomous flag is not set, the prefix is the link-local prefix, or if the preferred lifetime is greater than the valid life (a node may wish to log a system management error in this last case). If the prefix advertised does not match the prefix of an address already in the list, and the Valid Lifetime is not 0, form an address (and add it to the list) by combining the advertised prefix with the link's interface identifier. If the sum of the prefix length and interface identifier length does not equal 128 bits, the Prefix Information option MUST be ignored.

Test Setup: Initialize all devices on Link B. Common Test Setup 1.1 is performed. Initialize the HUT after each part. Prefix "X" and Prefix "Y" must not be included in the HUT's prefix list.

Router Advertisement A

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: Multicast Address
Router Advertisement Router Lifetime: 20 seconds Reachable Time: 10 seconds Retransmit Interval: 1 second
Prefix Option "on-link" (L) flag: 1 Valid Lifetime: 20 seconds

Preferred Lifetime: 20 seconds
Prefix: TR1's Global Prefix "X"

Procedure:

Part A: Router Advertisement with multiple Prefix Options

1. TR1 transmits a Router Advertisement with the Autonomous flag set, NextHop=255, and multiple prefix options, Prefix "X" with a valid lifetime of 20s and Prefix "Y" with a valid lifetime of 40s.
2. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X". Observe the packets on Link B.
3. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "Y". Observe the packets on Link B.
4. Wait for 21s so the lifetime expires for Prefix "X".
5. Repeat Step 2.
6. Wait for 20s so the lifetime expires for Prefix "Y".
7. Repeat Step 3.

Part B: Autonomous Flag not set

8. TR1 transmits a Router Advertisement A with the Autonomous flag not set.
9. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
10. Observe the packets on Link B.

Part C: prefix is set to link-local prefix

11. TR1 transmits Router Advertisement A with the prefix set the link-local prefix.
12. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
13. Observe the packets on Link B.

Part D: preferred lifetime > valid lifetime

14. TR1 transmits Router Advertisement A with the preferred lifetime set to 30 seconds.
15. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
16. Observe the packets on Link B.

Part E: prefix length > 128 bits

17. The HUT must have an interface identifier of length greater than zero. TR1 transmits Router Advertisement A with a Prefix Length set to 128.
18. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
19. Observe the packets on Link B.

Part F: prefix length < 64 bits

20. The HUT must have an interface identifier of length greater than zero. TR1 transmits Router Advertisement A with a Prefix Length set to zero.
21. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
22. Observe the packets on Link B.

Part G: (64 bits < prefix length < 128 bits)

23. The HUT must have an interface identifier of length greater than zero. TR1 transmits Router

Advertisement A with a Prefix Length set to 120.

24. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
25. Observe the packets on Link B.

Part H: Valid Lifetime is zero

26. TR1 transmits Router Advertisement A with the Valid Lifetime set to zero.
27. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
28. Observe the packets on Link B.

Part I: Invalid RA with Hop Limit 254

29. TR1 transmits Router Advertisement A with a Hop Limit set to 254.
30. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
31. Observe the packets on Link B.

Observable Results:

- *Part A*

  Step 1: The HUT should process the Prefix Information Options and form an address for each prefix.
  Step 2: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix "X".
  Step 3: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix "Y".
  Step 4: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix "X".
  Step 7: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix "Y".
- *Part B*

  Step 8: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix "X".
  Step 10: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix "X".
- *Part C*

  Step 11: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix "X".
  Step 13: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix "X".
- *Part D*

  Step 14: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix "X".
  Step 16: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix "X".
- *Part E*

  Step 17: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix "X".
  Step 19: The HUT must NOT transmit a Solicited NA for its autoconfigured global address

with Prefix “X”.

- *Part F*

  Step 20: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix “X”.
  Step 22: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
- *Part G*

  Step 23: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix “X”.
  Step 25: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
- *Part H*

  Step 26: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix “X”.
  Step 28: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
- *Part I*

  Step 29: The HUT should silently ignore the Prefix Information Option and not form an address using Prefix “X”.
  Step 31: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.

Possible Problems:

- None.

Test v6LC.3.2.4: Prefix-Information Option Processing, Lifetime (Hosts Only)

Purpose: Verify that a host properly updates its Address List upon receipt of Prefix Information Options.

References:

- [ADDRCONF] – Section 5.5.3

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The Prefix-Information option in the Router Advertisement is checked to see if the advertised prefix matches the prefix of an autoconfigured address in the list of addresses associated with the interface, the specific action to perform depends on the valid Lifetime in the received advertisement and the Lifetime associated with the previously autoconfigured address which is referred to as the StoredLifetime in this test.

If the received Lifetime is greater than 2 hours or greater than the StoredLiftetime, the stored Lifetime of the corresponding address is updated.

If the StoredLifetime is less than or equal to 2 hours and the received Lifetime is less than or equal to StoredLifetime, ignore the prefix, unless the Router Advertisement from which this Prefix Information option was obtained has been authenticated.

Otherwise, reset the StoredLifetime in the corresponding address to two hours.

Test Setup: Initialize the HUT before each part. Common Test Setup 1.1 is performed.

Router Advertisement A

IPv6 Header Next Header: 58 Source Address: TR1's Link-local Address Destination Address: Multicast Address
Router Advertisement Router Lifetime: 60 seconds Reachable Time: 600 seconds Retransmit Interval: 1 second
Prefix Option "on-link" (L) flag: 1 Valid Lifetime: 20 seconds Preferred Lifetime: 20 seconds Prefix: Global Prefix "X"

Procedure:

Part A: Prefix Lifetime greater than Stored Lifetime

1. TR1 transmits Router Advertisement A with a Valid Lifetime of 30 seconds.
2. Wait 10 seconds.
3. TR1 transmits a Router Advertisement with a prefix of TR1's Global Prefix and a Valid Lifetime of 60 seconds.
4. Wait 25 seconds.
5. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
6. Observe packets transmitted by the HUT.

Part B: Prefix Lifetime greater than 2 hours

7. TR1 transmits Router Advertisement A with a Valid Lifetime of 3hrs.
8. TR1 transmits a Router Advertisement with a prefix of TR1's Global Prefix and a Valid Lifetime of 2hrs 30s.
9. Wait 2hrs 45 seconds.
10. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
11. Observe packets transmitted by the HUT. (Should timeout the prefix)

Part C: Prefix Lifetime less than the Stored Lifetime and the Stored Lifetime is less than 2 hours

12. TR1 transmits Router Advertisement A with a Valid Lifetime of 60 seconds.
13. TR1 transmits a Router Advertisement with a prefix of TR1's Global Prefix and a Valid Lifetime of 30 seconds.
14. Wait 35 seconds.
15. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
16. Observe packets transmitted by the HUT.

Part D: Prefix Lifetime less than 2 hours and the Stored Lifetime is greater than 2 hours

17. TR1 transmits Router Advertisement A with a Valid Lifetime of 2hrs 30s.
18. TR1 transmits a Router Advertisement with a prefix of TR1's Global Prefix and a Valid Lifetime of 10 seconds.
19. Wait 11 seconds.
20. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X". Observe packets transmitted by the HUT.
21. Wait 2hrs 15 second
22. Configure TR1 to transmit a NS message for address resolution with the target address set to the HUT's global address for Prefix "X".
23. Observe packets transmitted by the HUT.

Observable Results:

- *Part A*

  Step 6: The HUT must update it's Stored Lifetime and must not timeout Prefix "X' after 30 seconds. The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix "X".

- *Part B*
  Step 11: The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
- *Part C*
  Step 16: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
- *Part D*
  Step 21: The HUT must transmit a Solicited NA for its autoconfigured global address with Prefix “X”.
  Step 23: The StoredLifetime should time out the global Prefix “X”. The HUT must NOT transmit a Solicited NA for its autoconfigured global address with Prefix “X”.

Possible Problems:

- None.

Section 4: RFC 1981

Scope

The following tests cover the Path MTU Discovery for IP version 6, Request For Comments 1981. The Path MTU Discovery protocol is a technique to dynamically discover the PMTU of a path. The basic idea is that a source node initially assumes that the PMTU of a path is the (known) MTU is the first hop in the path. If any of the packets sent on the path are too large to be forwarded by some node along the path, that node will discard them and return ICMPv6 Packet Too Big messages. Upon receipt of such a message, the source node reduces its assumed PMTU for the path based on the MTU of the constricting hop as reported in the Packet Too Big message. The Path MTU Discovery process ends when the nodes's estimate of the PMTU is less than or equal to the actual PMTU.

Overview

These tests are designed to verify the readiness of an IPv6 implementation vis-à-vis the Path MTU Discovery IPv6 specification.

Default Packets

Router Advertisement

IPv6 Header Source Address: TR1's Link-Local Address Destination Address: All-Nodes multicast address Next Header: 58
ICMPv6 Header Type: 134 Code: 0 M Bit (managed): 0 O Bit (other): 0 Router Lifetime: 20 seconds Reachable Time: 10 seconds Retrans Timer: 1 second
Prefix Option Type: 3 L Bit (on-link flag): 1 A Bit (addr conf): 1 Valid Lifetime: 20 seconds Preferred Lifetime: 20 seconds Prefix: link's prefix

Echo Request

IPv6 Header Payload Length: 1400 bytes Next Header: 58
ICMPv6 Header Type: 128 Code: 0

Packet Too Big message	Redirect message
IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address Destination Address: NUT's Link Local Address	IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address Destination Address: NUT's Link Local Address
ICMPv6 Header Type: 2 Code: 0 MTU: 1280	ICMPv6 Header Type: 137 Code: 0
Invoking Packet	Invoking Packet

*Note, if the media type is not Ethernet (MTU is not 1500), the payload in the Echo Request Packet should be adjusted so that it fits the default MTU.

*Flow id is consistent for all tests.

Test v6LC.4.1.1: Confirm Ping

Purpose: Verify that a node can reply to variable sized ICMP Echo Requests.

References:

- [ICMPv6] – Section 4.2
- [IPv6-SPEC] – Section 5
- This is a basic packet processing test.

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: The data received in an ICMPv6 Echo Request message MUST be returned entirely and unmodified in the ICMPv6 Echo Reply.

From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: ICMPv6 Echo Request 64 octets

1. TR1 forwards an Echo Request from TN2 to the NUT. The packet size of the Echo Request is 64 octets.
2. Observe the packets transmitted by the NUT.

Part B: ICMPv6 Echo Request 1280 octets

3. TR1 forwards an Echo Request from TN2 to the NUT. The packet size of the Echo Request is 1280 octets.
4. Observe the packets transmitted by the NUT.

Part C: ICMPv6 Echo Request 1500 octets

5. TR1 forwards an Echo Request from TN2 to the NUT. The packet size of the Echo Request is 1500 octets. (If the associated media type MTU default value is less than this, use that value instead.)
6. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT sent an Echo Reply to TR1 64 octets in packet size.
- *Part B*

  Step 4: The NUT should send an Echo Reply to TR1 1280 octets in packet size.

- *Part C*

  Step 6: The NUT should send an Echo Reply to TR1 1500 octets in packet size. (If the Echo Request was sent with a different size due to the associated media type default MTU value, then the Echo Reply sent should equal that size.)

Possible Problems:

- None.

Test v6LC.4.1.2: Stored PMTU

Purpose: Verify that a node can store Path MTU information for multiple destinations.

References:

- [PMTU] – Section 5.2
- This is a basic packet processing test.

Resource Requirements:

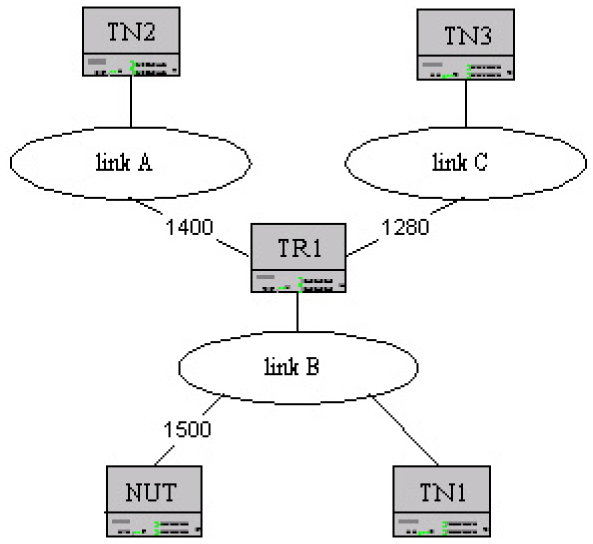
- Packet generator
  - Monitor to capture packets

Discussion: Ideally, a PMTU value should be associated with a specific path traversed by packets exchanged between the source and destination nodes. However, in most cases a node will not have enough information to completely and accurately identify such a path. Rather, a node MUST associate a PMTU value with some local representation of a path. It is left to the implementation to select the local representation of a path.

Minimally, an implementation could contain a single PMTU value to be used for all packets originated from that node.

Test Setup: The common cleanup procedure is performed after each part. The following setup is performed at the beginning of each test part. Refer to the diagram below:

1. If the NUT is a host, TR1 transmits a Router Advertisement to the all-nodes multicast address. The Router Advertisement includes a Prefix Advertisement with a global prefix and the L and A bits set. This should cause the NUT to add TR1 to its Default Router List, configure a global address, and compute Reachable Time. The Router and Prefix Lifetimes are long enough such that they do not expire during the test.
2. If the NUT is a router, configure a default route with TR1 as the next hop.
3. TR1 transmits an Echo Request to the NUT and responds to Neighbor Solicitations from the NUT. Wait for an Echo Reply from the NUT. This should cause the NUT to resolve the address of TR1 and create a Neighbor Cache entry for TR1 in state REACHABLE.
4. TR1's interface to TN2 on Link A has an MTU of 1400 octets.
5. TR1's interface to TN3 on Link C has an MTU of 1280 octets.
6. All other MTU values are 1500 octets or the default link MTU for the associated media type.



Procedure:

1. TN1 sends an Echo Request on-link to the NUT with packet size equal to 1500 octets.
2. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
3. TR1 forwards an Echo Request from TN3 to the NUT with packet size equal to 1500 octets.
4. Observe the packets transmitted by the NUT.
5. TR1 transmits a Packet Too Big message to the NUT for the Echo Reply to TN2, which contains an MTU field with a value of 1400.
6. TN1 sends an Echo Request on-link to the NUT with packet size equal to 1500 octets.
7. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
8. TR1 forwards an Echo Request from TN3 to the NUT with packet size equal to 1500 octets.
9. Observe the packets transmitted by the NUT.
10. TR1 transmits a Packet Too Big message to the NUT for the Echo Reply to TN3, which contains an MTU field with a value of 1280.
11. TN1 sends an Echo Request on-link to the NUT with packet size equal to 1500 octets.
12. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
13. TR1 forwards an Echo Request from TN3 to the NUT with packet size equal to 1500 octets.
14. Observe the packets transmitted by the NUT.

Observable Results:

Step 4: The NUT should send three Echo Replies, one to TN1, one to TN2, and one to TN3.
Step 9: The NUT should respond to the three Echo Requests. The Echo Replies to TN1 and TN3 should be no larger than 1500 octets. The NUT does not have to fragment these packets. The NUT should correctly fragment its Echo Reply to TN2 with each fragment no larger than 1400 octets. These fragments may be smaller. (See note in Possible Problems.)
Step 14: The NUT should respond to the three Echo Requests. The Echo Reply to TN1 should be no larger than 1500 octets. The NUT does not have to fragment this packet. The NUT should correctly fragment its Echo Reply to TN2 with each fragment no larger than 1400 octets. These fragments may be smaller. The NUT should correctly fragment its Echo Reply to TN3 with each fragment no larger than 1280 octets. (See note in Possible Problems.)

Possible Problems:

- The Node Under Test may choose to implement one PMTU for all destinations. In that case, each response in all parts should be no larger than 1280 octets.

Test v6LC.4.1.3: Non-zero ICMPv6 Code

Purpose: Verify that a node properly processes a Packet Too Big message with a non-zero ICMPv6 Code field.

References:

- [PMTU]
- [ICMPv6] – Section 3.2

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: The ICMPv6 Code field is set to zero by the sender and ignored by the receiver.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

1. TR1's link MTU on its interface to Link A (to TN2) is configured to be 1280 octets. This link MTU is smaller than the link MTU on its interface to Link B.

Procedure:

1. TR1 forwards an Echo Request from TN2 to the NUT.
2. Observe the packets transmitted by the NUT.
3. TR1 transmits a Packet Too Big message to the NUT, which contains an invalid ICMPv6 Code field value of 0xFF. The MTU field is set to 1280.
4. TR1 forwards an Echo Request from TN2 to the NUT.
5. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT should respond to the Echo Request using TR1 as a first hop.
Step 5: The NUT should correctly fragment its response to the Echo Request using TR1 as a first hop, indicating the NUT ignored the invalid ICMPv6 Code field and processed the Packet Too Big message. The fragmented packets must not be larger than 1280 octets in size.

Possible Problems:

- None.

Test v6LC.4.1.4: Reduce PMTU On-link

Purpose: Verify that a node properly processes a Packet Too Big message indicating a reduction in Path MTU for an on-link destination.

References:

- [PMTU] – Sections 3, 5.1

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Path MTU Discovery must be performed even in cases where a node "thinks" a destination is attached to the same link as itself. In a situation such as when a neighboring router acts as a proxy for some destination, the destination can appear to be directly connected but is in fact more than one hop away.

It is possible that a packetization layer is unable to change the message size that is transmitted. In this case, IPv6 handles large payloads by dividing them into fragments, with each fragment in a separate packet containing a Fragment Header.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

1. TR1 transmits a 1500 byte link-local Echo Request to the NUT.
2. Observe the packets transmitted by the NUT.
3. Even though TR1 is configured with a link MTU associated with its media type (1500 for Ethernet), TR1 transmits a Packet Too Big message to the NUT with an MTU of 1280.
4. TR1 transmits a 1500 byte link-local fragmented Echo Request to the NUT. The fragmented packets are no larger than 1280 octets in size.
5. Observe the packets transmitted by the NUT.
6. Re-boot the NUT. Refer to Common Test Setup 1.1
7. Repeat Steps 1 through 5, transmitting an on-link global Echo Request to the NUT for both Steps 1 and 4.

Observable Results:

Step 2: The NUT should respond to the Echo Request.
Step 5: The NUT should correctly fragment its response to the Echo Request, indicating the NUT processed the Packet Too Big message. The fragmented packets must not be larger than 1280 octets in size.

Possible Problems:

- None.

Test v6LC.4.1.5: Reduce PMTU Off-link

Purpose: Verify that a node properly reduces its estimate of the MTU for a path due to a Packet Too big message indicating a reduction in the Path MTU for a global destination.

References:

- [PMTU] – Sections 4, 5.1

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: When a node receives a Packet Too Big message, it MUST reduce its estimate of the PMTU for the relevant path, based on the value of the MTU field in the message. The precise behavior of a node in this circumstance is not specified, since different applications may have different requirements, and since different implementation architectures may favor different strategies.

It is possible that a packetization layer is unable to change the message size that is transmitted. In this case, IPv6 handles large payloads by dividing them into fragments, with each fragment in a separate packet containing a Fragment Header.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

1. TR1's link MTU on its interface to Link A (to TN2) is configured to be 1280 octets. This link MTU is smaller than the link MTU on its interface to Link B

Procedure:

1. TR1 forwards an Echo Request from TN2 to the NUT.
2. Observe the packets transmitted by the NUT.
3. TR1 transmits a Packet Too Big message to the NUT with an MTU field set to 1400 octets.
4. TR1 forwards an Echo Request from TN2 to the NUT with a packet size of 1500 octets.
5. Observe the packets transmitted by the NUT.
6. TR1 transmits another Packet Too Big message containing an MTU field set to 1280 octets.
7. TR1 forwards an Echo Request from TN2 to the NUT with a packet size of 1500 octets.
8. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT should respond to the Request using TR1 as the first hop.
Step 5: The NUT should correctly fragment its response to the Echo Request using TR1 as a first hop, indicating the NUT processed the Packet Too Big message. The fragmented packets must not be larger than 1400 octets in size.

Step 8: The NUT should correctly fragment its response to the Echo Request using TR1 as a first hop, indicating the NUT processed the Packet Too Big message. The fragmented packets must not be larger than 1280 octets in size.

Possible Problems:

- None.

Test v6LC.4.1.6: Receiving MTU Below IPv6 Minimum Link MTU

Purpose: Verify that a node does not reduce its estimate of the Path MTU below the IPv6 minimum link MTU.

References:

- [PMTU] – Section 4
- [IPv6-SPEC] – Section 5

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: A node MUST NOT reduce its estimate of the Path MTU below the IPv6 minimum link MTU. Note: A node may receive a Packet Too Big message reporting a next-hop MTU that is less than the IPv6 minimum link MTU. In that case, the node is not required to reduce the size of the subsequent packets sent on the path to less than the IPv6 minimum link MTU, but rather must include a Fragment header in those packets.

IPv6 requires that every link in the Internet have a MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: MTU equal to 0x0

1. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1280 octets.
2. Observe the packets transmitted by the NUT.
3. TR1 transmits a Packet Too Big message to the NUT, which contains an MTU field of 0x0.
4. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1280 octets.
5. Observe the packets transmitted by the NUT.

Part B: MTU equal to 512

6. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1280 octets.
7. Observe the packets transmitted by the NUT.
8. TR1 transmits a Packet Too Big message to the NUT, which has a MTU field of 512 octets.
9. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1280 octets.
10. Observe the packets transmitted by the NUT.

Observable Results:

- *Parts A and B*

  Steps 2 and 7: The NUT should respond to the Echo Request using TR1 as a first hop.
  Step 5 and 10: The NUT should respond to the Echo Request, and must not reduce the size of packets to below the IPv6 minimum link MTU. Instead, it must include a Fragment Header in the Echo Reply packet. The Echo Reply packet size should still be 1280 octets.

Possible Problems:

- None.

Test v6LC.4.1.7: Increase Estimate

Purpose: Verify that a node does not increase its estimate of the MTU for a path due to a Packet Too Big message.

References:

- [PMTU] – Section 4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: A node MUST NOT increase its estimate of the Path MTU in response to the contents of a Packet Too Big message. A message purporting to announce an increase in the Path MTU might be a stale packet that has been floating around in the network, a false packet injected as part of a denial-of-service attack, or the result of having multiple paths to the destination, each with a different PMTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: MTU increase

1. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
2. Observe the packets transmitted by the NUT.
3. TR1 transmits a Packet Too Big message to the NUT. The MTU field is 1304 octets.
4. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
5. Observe the packets transmitted by the HUT.
6. TR1 transmits a Packet Too Big message to the NUT. The MTU field is 1500 octets.
7. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1500 octets.
8. Observe the packets transmitted by the HUT.

Part B: MTU equal to 0x1FFFFFFF

9. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
10. Observe the packets transmitted by the HUT.
11. TR1 transmits a Packet Too Big message to the NUT. The MTU field is 1304 octets.
12. TR1 forwards an Echo Request from TN2 to the NUT with packet size equal to 1500 octets.
13. Observe the packets transmitted by the HUT.
14. TR1 transmits a Packet Too Big message to the NUT. The MTU field of 0x1FFFFFFF.
15. TR1 forwards an Echo Request from TN2 to the NUT. The packet size is 1500 octets.
16. Observe the packets transmitted by the NUT.

Observable Results:

- *Parts A and B*

Step 2 and 10: The NUT should respond to the Echo Request using TR1 as a first hop.
Step 5 and 13: The NUT should fragment the response to the Echo Request using TR1 as a first hop, indicating the NUT processed the Packet Too Big message.
Step 8 and 16: The NUT must correctly fragment the response to the Echo Request using TR1 as a first hop so the packet size is equal to or under 1304 octets. The NUT should not process the second Packet Too Big message indicating an increase in the PMTU.

Possible Problems:

- None.

Test v6LC.4.1.8: Router Advertisement with MTU Option (Hosts Only)

Purpose: Verify that a host properly processes a Router Advertisement with an MTU option and reduces its estimate.

References:

- [PMTU] Section 2
- [ND] – Sections 4.2 and 6.3.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: The receipt of a Router Advertisement MUST NOT invalidate all information received in a previous advertisement or from another source. However, when received information for a specific parameter (e.g., Link MTU) or option (e.g., Lifetime on a specific Prefix) differs from information received earlier, and the parameter/option can only have one value, the most recently received information is authoritative.

If the MTU option is present, hosts SHOULD copy the option's value into LinkMTU so long as the value is greater than or equal to the minimum link MTU [IPv6-SPEC] and does not exceed the default LinkMTU value specified in the link type specific document.

Path MTU is defined as the minimum link MTU of all links in a path between source and destination.

Test Setup: Refer to Common Test Setup 1.1. Only the common cleanup procedure is performed after each part.

Procedure:

1. TR1 forwards an Echo Request from TN2 to the HUT with packet size equal to 1500 octets.
2. Observe the packets transmitted by the HUT.
3. TR1 transmits a Router Advertisement with an MTU option set to 1280 to the all-nodes multicast address.
4. TR1 forwards a fragmented Echo Request from TN2 to the HUT with reassembled packet size equal to 1500 octets.
5. Observe the packets transmitted by the HUT.

Observable Results:

Step 2: The HUT should reply to the Request. The HUT does not have to fragment the reply.
Step 5: The HUT should update its Link MTU for TR1 to 1280 octets. The HUT should correctly fragment the response to the Echo Request, indicating the HUT adjusted its estimate of

the Path MTU to the new Link MTU for its first hop (also the destination). The fragmented packets must not be larger than 1280 octets in size.

Possible Problems:

- None.

Test v6LC.4.1.9: Checking For Increase in PMTU

Purpose: Verify that a node waits the proper amount of time to check for PMTU increases.

References:

- [PMTU] Section 4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: While a node MUST detect decreases in PMTU as fast as possible, it also MAY detect increases as well. Checking for increases requires sending a packet larger than the current PMTU estimate. If a Packet Too Big Message is not received, the PMTU must have increased to at least the size of the transmitted packet. Because checking for increases requires sending packets larger than the current estimated PMTU, and the likelihood is that the PMTU will not have increased, it MUST be done at infrequent intervals. Attempts to detect increases MUST NOT be done less than 5 minutes after a Packet Too Big Message has been received for the given path. The recommended interval is twice the minimum value (10 minutes).

Test Setup: Refer to Common Test Setup 1.1. Only the common cleanup procedure is performed after each part.

Procedure:

1. TR1 forwards an Echo Request from TN2 to the NUT.
2. Observe the packets transmitted by the NUT.
3. TR1 transmits a Packet Too Big message to the NUT. The MTU field is 1304 octets.
4. TR1 forwards an Echo Request from TN2 to the NUT.
5. Observe the packets transmitted by the NUT.
6. TR1 forwards an Echo Request from TN2 every 30 seconds for 5 minutes after the Packet Too Big Message was sent.
7. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT should respond to the Echo Request.
Step 5: The NUT should correctly fragment the response to the Echo Request, indicating it processed the Packet Too Big Message from TR1. The fragmented packets must not be larger than 1304 octets in size.

Step 7: The NUT must not transmit any packets larger than 1304 octets for 5 minutes from the time it received the Packet Too Big Message from TR1 in step 3.

Possible Problems:

- None.

Test v6LC.4.1.10: Multicast Destination – One Router

Purpose: Verify that a node properly chooses the PMTU for multicast destinations.

References:

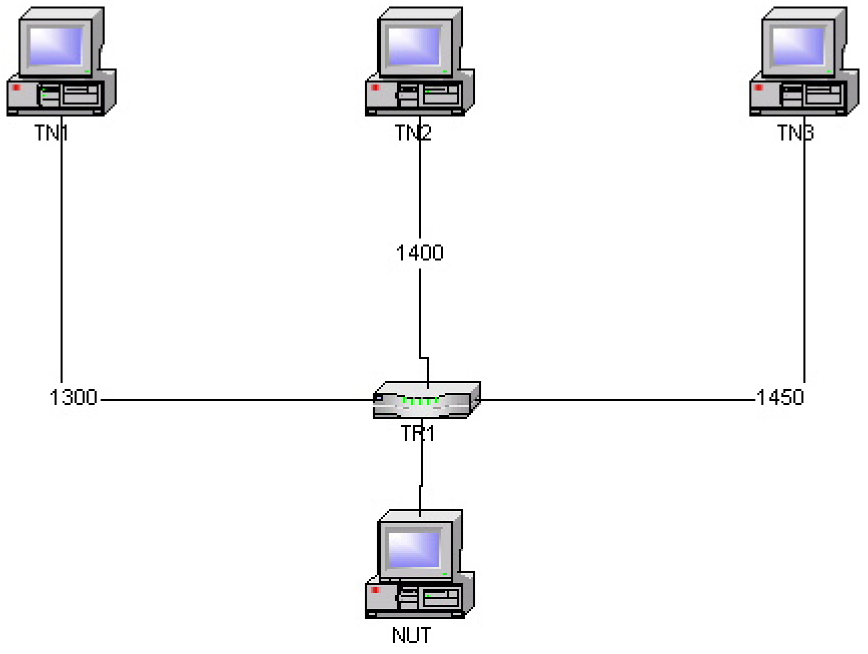
- [PMTU] Section 3

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Path MTU supports multicast as well as unicast destinations. When sending to multicast destinations, copies of a packet may traverse multiple paths to different nodes. Each path may have a different PMTU, and therefore a single multicast packet could cause multiple Packet Too Big Messages, with each reporting a different next-hop MTU. The minimum PMTU across all paths determines the size of packets to be sent to the multicast destination.

Test Setup: Only the common cleanup procedure is performed after each part.



1. TR1's Link MTU on its interface to TN1 is configured to be 1300 octets.
2. TR1's Link MTU on its interface to TN2 is configured to be 1400 octets.
3. TR1's Link MTU on its interface to TN3 is configured to be 1450 octets.
4. All other Link MTU's are set to the default for the associated media type.

5. TN1, TN2, and TN3 are all Listeners for the multicast group FF1E::1:2.

Procedure:

1. Transmit an ICMPv6 Echo Request from the NUT with packet size equal to 1500 octets and a destination to the multicast address of FF1E::1:2.
2. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1450.
3. Transmit the same packet as in Step 1 from the NUT.
4. Observe the packets transmitted by the NUT.
5. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1400.
6. Transmit the same packet as in Step 1 from the NUT.
7. Observe the packets transmitted by the NUT.
8. Transmit an ICMPv6 Echo Request from the NUT with packet size equal to 1400 octets and a destination to the multicast address of FF1E::1:2.
9. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1300.
10. Transmit the same packet as in Step 8 from the NUT.
11. Observe the packets transmitted by the NUT.
12. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1350.
13. Transmit the same packet as in Step 8 from the NUT.
14. Observe the packets transmitted by the NUT.

Observable Results:

Step 4: The NUT should correctly fragment its Echo Request to the multicast address of FF1E::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1450 octets in size.
Step 7: The NUT should correctly fragment its Echo Request to the multicast address of FF1E::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1400 octets in size.
Step 11: The NUT should correctly fragment its Echo Request to the multicast address of FF1E::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1300 octets in size.
Step 14: The NUT should correctly fragment its Echo Request to the multicast address of FF1E::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1300 octets in size.

Possible Problems:

- If the NUT is a “passive node”, it does not need to send an ICMPv6 Echo Request and may omit this test. This test must be performed if the NUT is a “non-passive node”, and is required to transmit an ICMPv6 Echo Request.

Test v6LC.4.1.11: Multicast Destination – Two Routers

Purpose: Verify that a node properly chooses the PMTU for multicast destinations when receiving PTB messages from more than one router.

References:

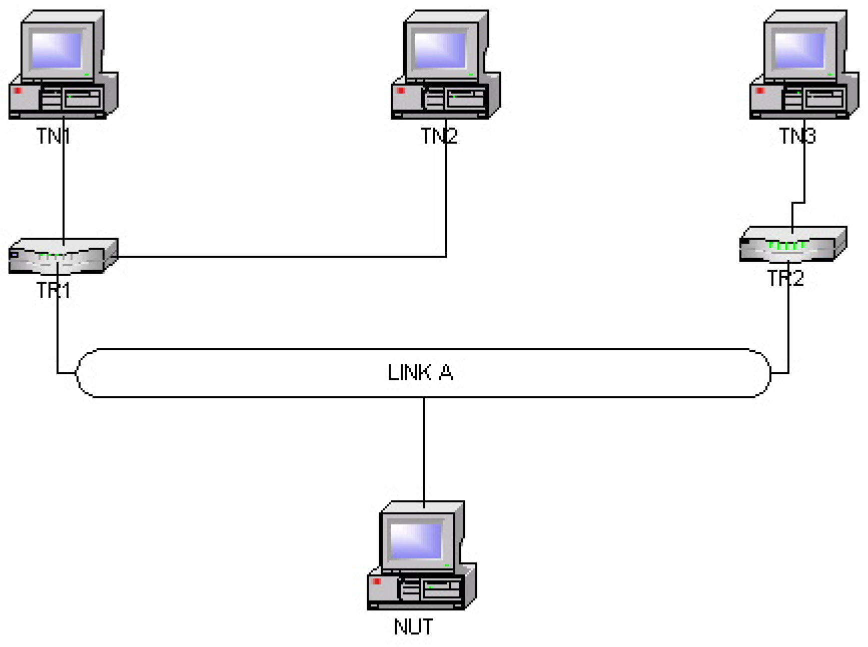
- [PMTU] Section 3

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Path MTU supports multicast as well as unicast destinations. When sending to multicast destinations, copies of a packet may traverse multiple paths to different nodes. Each path may have a different PMTU, and therefore a single multicast packet could cause multiple Packet Too Big Messages, with each reporting a different next-hop MTU. The minimum PMTU across all paths determines the size of packets to be sent to the multicast destination.

Test Setup: Only the common cleanup procedure is performed after each part.



1. All Link MTU’s are set to the default for the associated media type.
2. TN1, TN2, and TN3 are all Listeners for the multicast group FFE1::1:2.

Procedure:

1. Transmit an ICMPv6 Echo Request from the NUT with packet size equal to 1500 octets and a destination to the multicast address of FFE1::1:2.
2. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1480.
3. Transmit the same packet as in Step 1 from the NUT.
4. Observe the packets transmitted by the NUT.
5. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1440.
6. Transmit the same packet as in Step 1 from the NUT.
7. Observe the packets transmitted by the NUT.
8. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1400.
9. TR2 transmits a Packet Too Big Message to the NUT including an MTU field of 1360.
10. Transmit the same packet as in Step 1 from the NUT.
11. Observe the packets transmitted by the NUT.
12. TR1 transmits a Packet Too Big Message to the NUT including an MTU field of 1280.
13. TR2 transmits a Packet Too Big Message to the NUT including an MTU field of 1320.
14. Transmit the same packet as in Step 1 from the NUT.
15. Observe the packets transmitted by the NUT.

Observable Results:

Step 4: The NUT should correctly fragment its Echo Request to the multicast address of FFE1::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1480 octets in size.
Step 7: The NUT should correctly fragment its Echo Request to the multicast address of FFE1::1:2, indicating it processed the Packet Too Big Messages from TR1. The fragmented packets must not be larger than 1440 octets in size.
Step 11: The NUT should correctly fragment its Echo Request to the multicast address of FFE1::1:2, indicating it processed the Packet Too Big Messages from TR1 and TR2. The fragmented packets must not be larger than 1360 octets in size.
Step 15: The NUT should correctly fragment its Echo Request to the multicast address of FFE1::1:2, indicating it processed the Packet Too Big Messages from TR1 and TR2. The fragmented packets must not be larger than 1280 octets in size.

Possible Problems:

- If the NUT is a “passive node”, it does not need to send an ICMPv6 Echo Request and may omit this test. This test must be performed if the NUT is a “non-passive node”, and is required to transmit an ICMPv6 Echo Request.

Section 5: RFC 2463

Scope

The following tests cover the Internet Control Message Protocol for IP version 6, Request For Comments 2463.

Overview

These tests are designed to verify conformance with the Internet Control Message Protocol for the Internet Protocol Version 6 Specification.

Default Packets

Router Advertisement

IPv6 Header Source Address: TR1's Link-Local Address Destination Address: All-Nodes multicast address Next Header: 58
ICMPv6 Header Type: 134 Code: 0 M Bit (managed): 0 O Bit (other): 0 Router Lifetime: 90 seconds Reachable Time: 10 seconds Retrans Timer: 1 second
Prefix Option Type: 3 L Bit (on-link flag): 1 A Bit (addr conf): 1 Valid Lifetime: 90 seconds Preferred Lifetime: 90 seconds Prefix: link's prefix

Echo Request
(Packet Size 1500 bytes)

IPv6 Header Next Header: 58
ICMPv6 Header Type: 128 Code: 0

Packet Too Big message	Redirect message
IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address Destination Address: NUT's Link Local Address	IPv6 Header Next Header: 58 Source Address: TR1's Link Local Address Destination Address: NUT's Link Local Address
ICMPv6 Header Type: 2 Code: 0 MTU: 1280	ICMPv6 Header Type: 137 Code: 0
Invoking Packet	Invoking Packet

*Note: If the media type is not Ethernet (MTU is not 1500), the payload in the Echo Request Packet should be adjusted so that it fits the default MTU.

Link MTU Configuration

*Note: Some of these tests require the Node Under Test to configure link MTU on an interface. If this is not possible, the node may use a v6 over v4 tunnel (mtu = 1480), or a v6 over v6 tunnel (mtu = 1460).

Test v6LC.5.1.1: Transmitting Echo Requests

Purpose: Verify that a node properly transmits ICMPv6 Echo Requests.

References:

- [ICMPv6] – Section 2.2, 4.1

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Every node MUST implement an ICMPv6 Echo Responder function that receives Echo Requests and sends corresponding Echo Replies. A node SHOULD also implement an application-layer interface for sending Echo Requests and receiving Echo Replies, for diagnostic purposes.

The Destination Address of an ICMPv6 Echo Request message is any legal IPv6 address.

The Identifier field is an identifier to aid in matching Echo Replies to this Echo Request. This may be zero.

The Sequence Number field is a sequence number to aid in matching Echo Replies to this Echo Request. This may be zero.

The checksum is the 16-bit one's complement of the one's complement sum of the entire ICMPv6 message starting with the ICMPv6 message type field, prepended with a "pseudo-header" of IPv6 header fields, as specified in [IPv6-SPEC] section 8.1.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

1. Use Ping (or any available application for sending Echo Requests) to send an Echo Request from the NUT to TN1's Link-Local address.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must send an Echo Request to TN1. The Destination Address of the Packet must be same as TN1's Link-Local Address. The checksum must be valid. The Type field must be equal to 128 and the Code field must be equal to 0.

Possible Problems:

- If the NUT is a “passive node”, it does not need to send an ICMPv6 Echo Request and may omit this test. This test must be performed if the NUT is a “non-passive node”, and is required to transmit an ICMPv6 Echo Request.

Test v6LC.5.1.2: Replying to Echo Requests

Purpose: Verify that a node properly replies to ICMPv6 Echo Requests.

References:

- [ICMPv6] – Section 2.2, 4.2

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Every node MUST implement an ICMPv6 Echo Responder function that receives Echo Requests and sends corresponding Echo Replies. An Echo Reply SHOULD be sent in response to an Echo Request message sent to an IPv6 multicast address.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

If the message is a response to a message sent to a multicast or anycast group in which the node is a member, the Source Address of the reply MUST be a unicast address belonging to the interface on which the multicast or anycast packet was received.

The Destination Address of an ICMPv6 Echo Reply message is copied from the Source Address Field of the invoking Echo Request packet.

The checksum is the 16-bit one's complement of the one's complement sum of the entire ICMPv6 message starting with the ICMPv6 message type field, prepended with a "pseudo-header" of IPv6 header fields, as specified in [IPv6-SPEC] section 8.1.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: Request sent to Link-Local address

1. TN1 transmits an ICMPv6 Echo Request to the NUT's Link-Local address. The source address is TN1's Link-Local address.
2. Observe the packets transmitted by the NUT.

Part B: Request sent to global address

3. TN1 transmits an ICMPv6 Echo Request to the NUT's Global address. The source address is TN1's Global address.
4. Observe the packets transmitted by the NUT.

Part C: Request sent to multicast address

5. TN1 transmits an ICMPv6 Echo Request to the All-Nodes Link-Local Scope Multicast address (FF02::1). The source address is TN1's Link-Local address.
6. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must send an Echo Reply to TN1. The Source Address of the Packet must be same as the Link-Local Destination Address of TN1's Echo Request packet, while the Destination Address must be the same as the Link-Local Source Address of TN1's Echo Request packet. The NUT must send an Echo Reply to TN1 with a valid checksum.
- *Part B*

  Step 4: The NUT must send an Echo Reply to TN1. The Source Address of the Packet must be same as the Global Destination Address of TN1's Echo Request packet, while the Destination Address must be the same as the Global Source Address of TN1's Echo Request packet. The NUT must send an Echo Reply to TN1 with a valid checksum.
- *Part C*

  Step 6: The NUT should send an Echo Reply to TN1. The Source Address of the Packet must be one of the NUT's unicast addresses belonging to the interface on which the Echo Request was received. This could be either a Link-Local or Global address. The Destination Address must be TN1's local address Echo Request packet. The NUT must send an Echo Reply to TN1 with a valid checksum.

Possible Problems:

- None.

Test v6LC.5.1.3: Destination Unreachable Message Generation

Purpose: Verify that a node properly generates Destination Unreachable Messages.

References:

- [ICMPv6] – Section 2.2, 3.1, 2.4

Resource Requirements:

- Packet generator
- Monitor to capture packets

Discussion: An ICMPv6 Destination Unreachable Message SHOULD be generated by a router or by the IPv6 layer in the originating node in response to a packet that cannot be delivered to a destination for reasons other than congestion.

Code	Description
0	Lack of matching entry in the routing table
1	Administratively prohibited communication with destination
3	Address unreachable
4	Transport protocol has no listener on port (and protocol has no other means to inform sender)

If the message is a response to a message sent to an address that does not belong to the node, the Source Address should be that unicast address belonging to the node that will be most helpful in diagnosing the error.

The Destination Address of an ICMPv6 Destination Unreachable message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

1. The Payload Length of the ICMP Request Default Packets is 64 bytes.

Procedure:

Part A: Route Unreachable – Routers Only

1. If the RUT has any default routes in its routing table, delete them.
2. TN1 transmits an ICMPv6 Echo Request to an off-link address with a prefix that does not exist.

3. Observe the packets transmitted by the RUT.

Part B: Address Unreachable – Routers Only

4. TN1 transmits an ICMPv6 Echo Request to an on-link address that does not exist. The prefix should be set to the prefix assigned by the RUT.
5. Observe the packets transmitted by the RUT.

Part C: Port Unreachable – Link-Local Address – All Nodes

6. Make sure the NUT is not listening on port 9000.
7. TN1 transmits a UDP Packet with the destination port field set to 9000. The source address is TN1's Link-Local address.
8. Observe the packets transmitted by the NUT.

Part D: Port Unreachable – Global Address – All Nodes

9. Make sure the NUT is not listening on port 9000.
10. TN1 transmits a UDP Packet with the destination port field set to 9000. The source address is TN1's Global address.
11. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 3: The RUT should send a Destination Unreachable Message to TN1. The Source Address of the Packet should be one of the RUT's unicast addresses, while the Destination Address should be the same as the Source Address in TN1's Echo Request packet. The Code field should be set to "0". The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
- *Part B*

  Step 5: The RUT should send a Destination Unreachable Message to TN1. The Source Address of the Packet should be one of the RUT's unicast addresses, while the Destination Address should be the same as the Source Address in TN1's Echo Request packet. The Code field should be set to "3". The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
- *Part C*

  Step 8: The NUT should send a Destination Unreachable Message to TN1. The Source Address of the Packet should be one of the NUT's unicast addresses, while the Destination Address should be the same as the Link-Local Source Address in TN1's packet. The Code field should be set to "4". The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
- *Part D*

  Step 11: The NUT should send a Destination Unreachable Message to TN1. The Source Address of the Packet should be one of the NUT's unicast addresses, while the Destination Address should be the same as the Global Source Address in TN1's packet. The Code field should be set to "4". The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

Possible Problems:

- None.

Test v6LC.5.1.4: Packet Too Big Message Generation (Routers Only)

Purpose: Verify that a router properly generates Packet Too Big Messages.

References:

- [PMTU] – Section 3.2
- [ICMPv6] – Section 2.2, 3.2, 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: A Packet Too Big message MUST be sent by a router in response to a packet that it cannot forward because the packet is larger than the MTU of the outgoing link. The MTU field of the Packet Too Big Message is set to the Maximum Transmission Unit of the next-hop link.

If the message is a response to a message sent to an address that does not belong to the node, the Source Address should be that unicast address belonging to the node that will be most helpful in diagnosing the error. For example, if the message is a response to a packet forwarding action that cannot complete successfully, the Source Address should be a unicast address belonging to the interface on which the packet forwarding failed.

The Destination Address of an ICMPv6 Packet Too Big message is copied from the Source Address Field of the invoking packet. Code is set to 0 by the sender.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

1. Enable on the RUT an interface to Link A (to TN2).
2. Configure the RUT with a link MTU equal to the IPv6 minimum link MTU (1280 octets) on its interface to Link A (to TN2). (See Note Below in Possible Problems).
3. Configure all other interfaces on the RUT with the default link MTU for its associated media type. The link MTU for RUT's interface to Link A should be smaller than its link MTU to Link B.

Procedure:

Part A: Unicast Destination

1. TN1 transmits an Echo Request to TN2 using the RUT as the first-hop with a packet size of 1500 octets.
2. Observe the packets transmitted by the RUT.

Part B: Multicast Destination

3. Configure a multicast routing protocol on the RUT.
4. TN2 is a Listener for the multicast group FF1E::1:2.
5. TN1 transmits an Echo Request to the FF1E::1:2 address with a packet size of 1500 octets.
6. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*

  Step 2: The RUT must transmit a Packet Too Big message to TN1, as it could not forward the Echo Request due to PMTU limitations.
  - The MTU field of Packet Too Big Message should be set to 1280.
  - The Source Address of the Packet should be one of the RUT's unicast addresses for its interface to Link A (to TN2).
  - The Destination Address should be the same as the Source Address in TN1's Echo Request packet. The Code field should be set to "0".
  - The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
- *Part B*

  Step 6: The RUT must transmit a Packet Too Big message to TN1, as it could not reply to the Echo Request due to PMTU limitations.
  - The MTU field of Packet Too Big Message should be set to 1280.
  - The Source Address of the Packet should be one of the RUT's unicast addresses for its interface to Link A (to TN2).
  - The Destination Address should be the same as the Source Address in TN1's Echo Request packet. The Code field should be set to "0".
  - The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

Possible Problems:

- Note: Some of these tests require the Node Under Test to configure link MTU on an interface. If this is not possible, the node may use a v6 over v4 tunnel (mtu = 1480), or a v6 over v6 tunnel (mtu = 1460). If the MTU is not configurable at all for the Node Under Test, this test may be omitted.
- Part B can be omitted if the RUT does not support Multicast Routing.

Test v6LC.5.1.5: Hop Limit Exceeded (Time Exceeded Generation) (Routers Only)

Purpose: Verify that a router properly generates Time Exceeded Messages the Hop Limit was exceeded in transit.

References:

- [ICMPv6] – Section 2.2, 3.3, 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: If a router receives a packet with a Hop Limit of zero, or a router decrements a packet's Hop Limit to zero, it MUST discard the packet and send an ICMPv6 Time Exceeded message with Code 0 to the source of the packet. This indicates either a routing loop or too small an initial Hop Limit value.

The unused field MUST be initialized to zero by the sender and ignored by the receiver.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

If the message is a response to a message sent to an address that does not belong to the node, the Source Address should be that unicast address belonging to the node that will be most helpful in diagnosing the error. For example, if the message is a response to a packet forwarding action that cannot complete successfully, the Source Address should be a unicast address belonging to the interface on which the packet forwarding failed.

The Destination Address of an ICMPv6 Time Exceeded message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

1. Enable on the RUT an interface to link A (to TN2).

Packet A (Echo Request)

IPv6 Header Payload Length: 64 bytes Next Header: 58 Hop Limit: 0
ICMPv6 Header Type: 128 Code: 0

Packet B (Echo Request)

IPv6 Header Payload Length: 64 bytes Next Header: 58 Hop Limit: 1
ICMPv6 Header Type: 128 Code: 0

Procedure:

Part A: Receive Hop Limit 0

1. TN1 transmits the Packet A Echo Request to TN2 with a first hop of the RUT.
2. Observe the packets transmitted by the RUT.

Part B: Decrement Hop Limit to 0

3. TN1 transmits the Packet B Echo Request to TN2 with a first hop of the RUT.
4. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*
  Step 2: The RUT must discard the ICMPv6 Echo Request from TN1. Therefore, it must not forward the Echo Request to TN2. The RUT should send a Time Exceeded Message to TN1 with a code field value of 0 (Hop Limit Exceeded in transit).
  - The unused field must be initialized to zero.
  - The Source Address of the Packet should be one of the RUT's unicast addresses used for packet forwarding.
  - The Destination Address should be the same as TN1's Source Address.
  - The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.
- *Part B*
  Step 4: The RUT must discard the ICMPv6 Echo Request from TN1. Therefore, it must not forward the Echo Request to TN2. The RUT should decrement the Hop Limit to 0 and send a Time Exceeded Message to TN1 with a code field value of 0 (Hop Limit Exceeded in transit).
  - The unused field must be initialized to zero.
  - The Source Address of the Packet should be one of the RUT's unicast addresses used for packet forwarding.

- The Destination Address should be the same as TN1’s Source Address.
- The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

Possible Problems:

- None.

Test v6LC.5.1.6: Erroneous Header Field (Parameter Problem Generation)

Purpose: Verify that a node properly generates Parameter Problem Messages for an Erroneous Header Field.

References:

- [IPv6-SPEC] – Section 4.5
- [ICMPv6] – Section 2.2, 3.4, 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: If an IPv6 node processing a packet finds a problem with a field in the IPv6 header or extensions headers such that it cannot complete processing the packet, it MUST discard the packet and SHOULD send an ICMPv6 Parameter Problem message to the packet's source, indicating the type and location of the problem.

A Code of 0 indicates an Erroneous Header Field was encountered.

If the length of a fragment, as derived from the fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is 1, then that fragment must be discarded and an ICMP Parameter Problem, Code 0 (Erroneous Header Field encountered), message should be sent to the source of the fragment, pointing to the Payload Length field of the fragment packet.

The pointer field identifies the octet of the original packet's header where the error was detected.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Packet A

IPv6 Header Next Header: 44 Payload Length: 37
Fragment Header Next Header: 58 Fragment Offset: 0 More Fragments flag: 1
ICMPv6 Echo Request Data Length: 5

Procedure:

1. TN1 transmits the Packet A Echo Request to the NUT. The Source Address of the Packet is set to TN1's Global address. The Destination Address of the packet is set to the NUT's Global address.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must discard the ICMPv6 Echo Request from TN1. Therefore, it must not send an Echo Reply. The NUT should send a Parameter Problem Message to TN1 with a code field value of 0 (Erroneous Header Field encountered) because the Payload Length is not a multiple of 8 octets.

- The Pointer Field should be 0x04 (offset of the Payload Length field).
- The Source Address of the Packet must be the same as the Global Destination Address of TN1's Echo Request packet.
- The Destination Address should be the same as the Global Source Address of TN1's Echo Request packet.
- The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

Possible Problems:

- None.

Test v6LC.5.1.7: Unrecognized Next Header (Parameter Problem Generation)

Purpose: Verify that a node properly generates Parameter Problem Messages when an Unrecognized Next Header type is encountered.

References:

- [ICMPv6] – Section 2.2, 3.4, 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: If an IPv6 node processing a packet finds a problem with a field in the IPv6 header or extensions headers such that it cannot complete processing the packet, it MUST discard the packet and SHOULD send an ICMPv6 Parameter Problem message to the packet's source, indicating the type and location of the problem.

A Code of 1 indicates an unrecognized Next Header type was encountered.

The pointer field identifies the octet of the original packet's header where the error was detected.

If the message is a response to a message sent to one of the node's unicast addresses, the Source Address of the reply MUST be that same address.

The Destination Address of an ICMPv6 Parameter Problem message is copied from the Source Address Field of the invoking packet.

Implementations MUST observe the following rule when processing ICMPv6 messages:
Every ICMPv6 Error message includes as much of the IPv6 offending (invoking) packet as will fit without making the error message packet exceed the minimum IPv6 MTU.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Packet A (Echo Request)

IPv6 Header Payload Length: 64 bytes Next Header: 60
Destination Options Header Next Header: 128 Header Ext. Length: 0 PadN Option
ICMPv6 Header Type: 128 Code: 0

Procedure:

1. TN1 transmits the Packet A Echo Request to the NUT. The Source Address of the Packet is set to TN1's Global address. The Destination Address of the packet is set to the NUT's Global address.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must discard the ICMPv6 Echo Request from TN1. Therefore, it must not send an Echo Reply. The NUT should send a Parameter Problem Message to TN1 with a code field value of 1 (Unrecognized Next Header type encountered).

- The Pointer Field should be 0x28 (offset of the Next Header field).
- The Source Address of the Packet must be the same as the Global Destination Address of TN1's Echo Request packet.
- The Destination Address should be the same as the Global Source Address of TN1's Echo Request packet.
- The invoking Echo Request packet included in the Error Message must not exceed minimum IPv6 MTU.

Possible Problems:

- None.

Test v6LC.5.1.8: Unknown Informational Message Type

Purpose: Verify that a node properly handles the reception of an ICMPv6 Packet with an Unknown Informational Message Type value.

References:

- [ICMPv6] – Section 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: Implementations MUST observe the following rule when processing ICMPv6 messages: If an ICMPv6 informational message of unknown type is received, it must be silently discarded.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

1. TN1 transmits an ICMPv6 Information Message with a type field value of 255 to the NUT.
2. Observe the packets transmitted by the NUT.

Observable Results:

Step 2: The NUT must silently discard the ICMPv6 Informational Message from TN1.

Possible Problems:

- None.

Test v6LC.5.1.9: Error Condition With ICMPv6 Error Message (Routers Only)

Purpose: Verify that a router properly handles the reception and processing of an ICMPv6 Error Message that invokes an error.

References:

- [ICMPv6] – Section 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: An ICMPv6 error message MUST NOT be sent as a result of receiving an ICMPv6 error message.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: Reception of Flawed Destination Unreachable Code 0 with Address Unreachable

1. TN1 transmits a Destination Unreachable Error Message for "No Route To Destination" to the RUT with the Destination Address set to an on-link address that does not exist.
2. Observe the packets transmitted by the RUT.

Part B: Reception of Flawed Destination Unreachable Code 3 with Hop Limit = 0

3. TN1 transmits a Destination Unreachable Error Message for "Address Unreachable" to the RUT with the Hop Limit set to Zero in the IPv6 header and with a Destination Address set to an off-link address.
4. Observe the packets transmitted by the RUT.

Part C: Reception of Flawed Time Exceeded Code 0 with No Route To Destination

5. TN1 transmits a Time Exceeded Error Message for "Hop Limit Exceeded in Transit" to the RUT with the Destination Address set to an off-link address that does not exist.
6. Observe the packets transmitted by the RUT.

Part D: Reception of Flawed Time Exceeded Code 1 with No Route To Destination

7. TN1 transmits a Time Exceeded Error Message for "Fragment Reassembly Time Exceeded" to the RUT with the Destination Address set to an off-link address that does not exist.
8. Observe the packets transmitted by the RUT.

Part E: Reception of Flawed Packet Too Big with Address Unreachable

9. TN1 transmits a Packet Too Big Error Message to the RUT with the Destination Address set to an on-link address that does not exist.
10. Observe the packets transmitted by the RUT.

Part F: Reception of Flawed Parameter Problem with Hop Limit = 0

11. TN1 transmits a Parameter Problem Error Message to the RUT with the Hop Limit set to Zero in the IPv6 header and with a Destination Address set to an off-link address.

12. Observe the packets transmitted by the RUT.

Observable Results:

- *Part A*

  Step 2: The RUT must not send a Destination Unreachable Error Message with Code 3 to TN1 when it receives a Destination Unreachable Message with Code 0 for which it cannot resolve a destination address.
- *Part B*

  Step 4: The RUT must not send a Time Exceeded message with Code 0 to TN1 when it receives a Destination Unreachable Message with Code 3 that contains a Hop Limit of 0.
- *Part C*

  Step 6: The RUT must not send a Destination Unreachable Error Message with code 0 to TN1 when it receives a Time Exceeded Message with Code 0 for which it cannot route.
- *Part D*

  Step 8: The RUT must not send a Destination Unreachable Error Message with code 0 to TN1 when it receives a Time Exceeded Message with Code 1 for which it cannot route.
- *Part E*

  Step 10: The RUT must not send a Destination Unreachable Error Message with code 3 to TN1 when it receives a Packet Too Big Message for which it cannot resolve a destination address.
- *Part F*

  Step 12: The RUT must not send a Time Exceeded Error Message with code 0 to TN1 when it receives a Parameter Problem Message that contains a Hop Limit of 0.

Possible Problems:

- None.

Test v6LC.5.1.10: Error Condition With Multicast Destination

Purpose: Verify that a node properly handles the reception of an error condition caused by a packet with a Multicast Destination Address.

References:

- [ICMPv6] – Section 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: An ICMPv6 error message MUST NOT be sent as a result of receiving a packet destined to an IPv6 multicast address. (There are two exceptions to this rule: (1) the Packet Too Big Message – Section 3.2 – to allow Path MTU discovery to work for IPv6 multicast, and (2) the Parameter Problem Message, Code 2 – Section 3.4 – reporting an unrecognized IPv6 option that has the Option Type highest-order two bits set to 10.)

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: UDP Port Unreachable

1. TN1 transmits a UDP packet on Link B with the Destination Address set to the all-nodes link-local multicast address. The destination port is set to 9000. (Make sure the NUT is not listening on port 9000.)
2. Observe the packets transmitted by the NUT.

Part B: Echo Request Reassembly Timeout

3. TN1 transmits an ICMPv6 Echo Request Fragment to the all-nodes link-local multicast address. The offset of the fragment is 0 (the first fragment) and the More Fragments Flag is set.
4. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*
  Step 2: The NUT must not send a Destination Unreachable Error Message to TN1 when it receives a UDP packet for an unreachable port.
- *Part B*
  Step 4: The NUT must not send a Time Exceeded Error Message to TN1 60 seconds after it receives the first fragment of an ICMPv6 Echo Request.

Possible Problems:

- None.

Test v6LC.5.1.11: Error Condition With Non-Unique Source - Unspecified

Purpose: Verify that a node properly handles the reception of an error condition caused by a packet with a source address that does not uniquely identify a single node.

References:

- [ICMPv6] – Section 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: An ICMPv6 error message MUST NOT be sent as a result of receiving a packet whose source address does not uniquely identify a single node -- e.g., the IPv6 Unspecified Address, an IPv6 multicast address, or an address known by the ICMP message sender to be an IPv6 anycast address.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: UDP Port Unreachable (Routers and Hosts)

1. TN1 transmits a UDP Packet to the NUT's Global address with a Source Address set to the unspecified address (::). The destination port is set to 9000. (Make sure the NUT is not listening on port 9000.)
2. Observe the packets transmitted by the NUT.

Part B: Echo Request Too Big (Routers Only)

3. Configure the RUT with a link MTU equal to the IPv6 minimum link MTU (1280 octets) on its interface to Link A (to TN2). (See Note Below in Possible Problems).
4. Enable the RUT's interface to Link A (to TN2).
5. Configure all other interfaces on the RUT with the default link MTU for its associated media type. The link MTU for RUT's interface to Link A should be smaller than its link MTU to Link B.
6. TN1 transmits an ICMPv6 Echo Request with a total message size of 1500 octets to TN2 with a first hop through the RUT. The Source Address is set to the unspecified address (::).
7. Observe the packets transmitted by the RUT.

Part C: Echo Request Reassembly Timeout (Routers and Hosts)

8. TN1 transmits an ICMPv6 Echo Request Fragment to the NUT. The offset of the fragment is 0 (the first fragment) and the More Fragments Flag is set. The Source Address is set to the unspecified address (::).
9. Observe the packets transmitted by the NUT.

Part D: Echo Request with Unknown Option in Destination Options (Routers and Hosts)

10. TN1 transmits an ICMPv6 Echo Request to the NUT. The Source Address is set to the unspecified address (::). It includes a Destination Options Header with the unrecognized Option of type 135. (Highest Order bits set to 10_b).
11. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*
  Step 2: The NUT must not send a Destination Unreachable Error Message to TN1 when it receives a UDP packet for an unreachable port.
- *Part B*
  Step 7: The RUT must not send a Packet Too Big Error Message to TN1 when it receives an ICMPv6 Echo Request that is too large for it to send on its outgoing interface.
- *Part C*
  Step 9: The NUT must not send a Time Exceeded Error Message to TN1 60 seconds after it receives the first fragment of an ICMPv6 Echo Request.
- *Part D*
  Step 11: The NUT must not send a Parameter Problem Error Message when it receives an ICMPv6 Echo Request with an unknown option with highest bits 10_b.

Possible Problems:

- Note: Some of these tests require the Node Under Test to configure link MTU on an interface. If this is not possible, the node may use a v6 over v4 tunnel (mtu = 1480), or a v6 over v6 tunnel (mtu = 1460). If the MTU is not configurable at all for the Node Under Test, this test may be omitted.

Test v6LC.5.1.12: Error Condition With Non-Unique Source - Multicast

Purpose: Verify that a node properly handles the reception of an error condition caused by a packet with a source address that does not uniquely identify a single node.

References:

- [ICMPv6] – Section 2.4

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: An ICMPv6 error message MUST NOT be sent as a result of receiving a packet whose source address does not uniquely identify a single node -- e.g., the IPv6 Unspecified Address, an IPv6 multicast address, or an address known by the ICMP message sender to be an IPv6 anycast address.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: UDP Port Unreachable (Routers and Hosts)

1. TN1 transmits a UDP Packet to the NUT's Global Address with a Source Address set to TN1's Solicited-Node Multicast address. The destination port is set to 9000. (Make sure the NUT is not listening on port 9000.)
2. Observe the packets transmitted by the NUT.

Part B: Echo Request Too Big (Routers Only)

3. Configure the RUT with a link MTU equal to the IPv6 minimum link MTU (1280 octets) on its interface to Link A (to TN2). (See Note Below in Possible Problems).
4. Enable the RUT's interface to Link A (to TN2).
5. Configure all other interfaces on the RUT with the default link MTU for its associated media type. The link MTU for RUT's interface to Link A should be smaller than its link MTU to Link B.
6. TN1 transmits an ICMPv6 Echo Request with a total message size of 1500 octets to TN2 with a first hop through the RUT. The Source Address is set to TN1's Solicited-Node Multicast address.
7. Observe the packets transmitted by the RUT.

Part C: Echo Request Reassembly Timeout (Routers and Hosts)

8. TN1 transmits an ICMPv6 Echo Request Fragment to the NUT. The offset of the fragment is 0 (the first fragment) and the More Fragments Flag is set. The Source Address is set to TN1's Solicited-Node Multicast address.
9. Observe the packets transmitted by the NUT.

Part D: Echo Request with Unknown Option in Destination Options (Routers and Hosts)

10. TN1 transmits an ICMPv6 Echo Request to the NUT. The Source Address is set to TN1's Solicited-Node Multicast address. It includes a Destination Options Header with the unrecognized Option of type 135. (Highest Order bits set to 10_b).
11. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*

  Step 2: The NUT must not send a Destination Unreachable Error Message to TN1 when it receives a UDP packet for an unreachable port.
- *Part B*

  Step 7: The RUT must not send a Packet Too Big Error Message to TN1 when it receives an ICMPv6 Echo Request that is too large for it to send on its outgoing interface.
- *Part C*

  Step 9: The NUT must not send a Time Exceeded Error Message to TN1 60 seconds after it receives the first fragment of an ICMPv6 Echo Request.
- *Part D*

  Step 11: The NUT must not send a Parameter Problem Error Message when it receives an ICMPv6 Echo Request with an unknown option with highest bits 10_b.

Possible Problems:

- Note: Some of these tests require the Node Under Test to configure link MTU on an interface. If this is not possible, the node may use a v6 over v4 tunnel (mtu = 1480), or a v6 over v6 tunnel (mtu = 1460). If the MTU is not configurable at all for the Node Under Test, this test may be omitted.

Test v6LC.5.1.13: Error Condition With Non-Unique Source – Anycast (Routers Only)

Purpose: Verify that a node properly handles the reception of an error condition caused by a packet with a source address that does not uniquely identify a single node.

References:

- [ICMPv6] – Section 2.4
- [ADDR] – Section 2.6.1

Resource Requirements:

- Packet generator
  - Monitor to capture packets

Discussion: An ICMPv6 error message MUST NOT be sent as a result of receiving a packet whose source address does not uniquely identify a single node -- e.g., the IPv6 Unspecified Address, an IPv6 multicast address, or an address known by the ICMP message sender to be an IPv6 anycast address.

The Subnet-Router anycast address is predefined. Its format is as follows:

```
|                         n bits                 |   128-n bits   |
+------------------------------------------------+----------------+
|                   subnet prefix                | 00000000000000 |
+------------------------------------------------+----------------+
```

The "subnet prefix" in an anycast address is the prefix which identifies a specific link. This anycast address is syntactically the same as a unicast address for an interface on the link with the interface identifier set to zero.

Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet. All routers are required to support the Subnet-Router anycast addresses for the subnets which they have interfaces.

Test Setup: Refer to Common Test Setup 1.1. The common setup is performed at the beginning of each test part. The common cleanup procedure is performed after each part.

Procedure:

Part A: UDP Port Unreachable

1. TR1 transmits a UDP Packet to the NUT's Global Address with a Source Address set to TR1's Subnet-Router Anycast Address. The destination port is set to 9000. (Make sure the NUT is not listening on port 9000.)
2. Observe the packets transmitted by the NUT.

Part B: Echo Request Too Big

3. Configure the RUT with a link MTU equal to the IPv6 minimum link MTU (1280 octets) on its interface to Link A (to TN2). (See Note Below in Possible Problems).
4. Enable the RUT's interface to Link A (to TN2).
5. Configure all other interfaces on the RUT with the default link MTU for its associated media type. The link MTU for RUT's interface to Link A should be smaller than its link MTU to Link B.
6. TN1 transmits an ICMPv6 Echo Request with a total message size of 1500 octets to TN2 with a first hop through the RUT. The Source Address is set to TR1's Subnet-Router Anycast Address. (Because the RUT has an address configured with TR1's prefix, TR1's Subnet-Router Anycast Address is also the RUT's.).
7. Observe the packets transmitted by the RUT.

Part C: Echo Request Reassembly Timeout

8. TN1 transmits an ICMPv6 Echo Request Fragment to the NUT. The offset of the fragment is 0 (the first fragment) and the More Fragments Flag is set. The Source Address is set to TR1's Subnet-Router Anycast Address.
9. Observe the packets transmitted by the NUT.

Part D: Echo Request with Unknown Option in Destination Options

10. TN1 transmits an ICMPv6 Echo Request to the NUT. The Source Address is set to TR1's Subnet-Router Anycast Address. It includes a Destination Options Header with the unrecognized Option of type 135. (Highest Order bits set to 10_b).
11. Observe the packets transmitted by the NUT.

Observable Results:

- *Part A*
  Step 2: The NUT must not send a Destination Unreachable Error Message to TN1 when it receives a UDP packet for an unreachable port.
- *Part B*
  Step 7: The RUT must not send a Packet Too Big Error Message to TN1 when it receives an ICMPv6 Echo Request that is too large for it to send on its outgoing interface.
- *Part C*
  Step 9: The NUT must not send a Time Exceeded Error Message to TN1 60 seconds after it receives the first fragment of an ICMPv6 Echo Request.
- *Part D*
  Step 11: The NUT must not send a Parameter Problem Error Message when it receives an ICMPv6 Echo Request with an unknown option with highest bits 10_b.

Possible Problems:

- Note: Some of these tests require the Node Under Test to configure link MTU on an interface. If this is not possible, the node may use a v6 over v4 tunnel (mtu = 1480), or a v6 over v6 tunnel (mtu = 1460). If the MTU is not configurable at all for the Node Under Test, this test may be omitted.