Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 20, 2014

W. Cerveny
Arbor Networks

October 17, 2013

Benchmarking Neighbor Discovery Problems
draft-cerveny-bmwg-ipv6-nd-02

Abstract

This document is a benchmarking instantiation of RFC 6583: "Operational Neighbor Discovery Problems" [RFC6583]. It describes a general testing procedure and measurements that can be performed to evaluate how the problems described in RFC 6583 may impact the functionality or performance of intermediate nodes.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 20, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document is a benchmarking instantiation of RFC 6583: "Operational Neighbor Discovery Problems" [RFC6583]. It describes a general testing procedure and measurements that can be performed to evaluate how the problems described in RFC 6583 may impact the functionality or performance of intermediate nodes.

2. Terminology

Intermediate Node A router, switch, firewall or any other device which separates end-nodes. The tests in this document can be completed with any intermediate node which maintains a neighbor cache, although not all measurements and performance characteristics may apply.

Neighbor Cache The neighbor cache is a database which correlates the link-layer address and the adjacent interface with an IPv6 address.

Neighbor Discovery See Section 1 of RFC 4861 [RFC4861]

Non-participating Network Network connected to DUT, for which nodes are neither active participants nor directly impacted by the test traffic.

Scanner Network The network from which the scanning tested is connected.

Scanning Interface The interface from which the scanning activity is conducted.

Target Network The network for which the scanning tests is targeted.

Target Network Destination Interface The interface that resides on the target network, which is primarily used to measure DUT performance while the scanning activity is occurring.

3. Overview of Relevant NDP and Intermediate Node Behavior

In a traditional network, an intermediate node must support a mapping between a connected node’s IP address and the connected node’s link-layer address and interface the node is connected to. With IPv4, this process is handled by ARP [RFC0826]. With IPv6, this process is handled by NDP and is documented in [RFC4861].With IPv6, when a packet arrives on one of an intermediate node’s interfaces and the destination address is determined to be reachable via an adjacent network:

1. The intermediate node first determines if the destination IPv6 address is present in its neighbor cache.

2. If the address is present in the neighbor cache, the intermediate node forwards the packet to the destination node using the appropriate link-layer address and interface.

3. If the destination IPv6 address is not in the intermediate node’s neighbor cache:

    1. An entry for the IPv6 address is added to the neighbor cache and the entry is marked "INCOMPLETE".

    2. The intermediate node sends a neighbor solicitation packet to the solicited-node multicast address on the interface considered on-link.

    3. If a solicited neighbor advertisement for the IPv6 address is received by the intermediate node, the neighbor cache entry is marked "REACHABLE" and remains in this state for 30 seconds.

    4. If a neighbor advertisement is not received, the intermediate node will continue sending neighbor solicitation packets every second until either a neighbor solicitation is received or the maximum number of solicitations has been sent. If a neighbor advertisement is not received in this period, the entry can be discarded.

There are two scenarios where a neighbor cache can grow to a very large size:

1. There are a large number of real nodes connected via an intermediate node’s interface and a large number of these nodes are sending and receiving traffic simultaneously.

2. There are a large number of addresses for which a scanning activity is occuring and no real node will respond to the neighbor solicitation. This scanning activity can be unintentional or malicious. In addition to maintaining the "INCOMPLETE" neighbor cache entry, the intermediate node must send a neighbor solicitation packet every second for the maximum number of socicitations. With today’s network link bandwidths, a scanning event could cause a lot of entries to be added to the neighbor cache and solicited for in the time that it takes for a neighbor cache entry to be discarded.

An intermediate node's neighbor cache is of a finite size and can only accommodate a specific number of entries, which can be limited by available memory or a preset operating system limit. If the maximum number of entries in a neighbor cache is reached, the intermediate node must either drop an existing entry to make space for the new entry or deny the new IP address to MAC address/ interface mapping with an entry in the neighbor cache. In an extreme case, the intermediate node's memory may become exhausted, causing the intermediate node to crash or begin paging memory.

At the core of the neighbor discovery problems presented in RFC 6583 [RFC6583], unintentional or malicious IPv6 traffic can transit the intermediate node that resembles an IP address scan similar to an IPv4-based network scan. Unlike IPv4 networks, an IPv6 end network is typically configured with a /64 address block, allowing for upwards of 2**64 addresses. When a network node attempts to scan all the addresses in a /64 address block directly attached to the intermediate node, it is possible to create a huge amount of state in the intermediate node's neighbor cache, which may stress processing or memory resources.

Section 7.1 of RFC 6583 recommends how intermediate nodes should behave when the neighbor cache is exceeded. Section 6 of RFC 6583 [RFC6583] recommends how damage from an IPv6 address scan may be mitigated. Section 6.2 of RFC 6583 [RFC6583] discusses queue tuning.

4. Test Setup

The network needs to minimally have two subnets: one from which the scanner(s) source their scanning activity and the other which is the target network of the address scans.

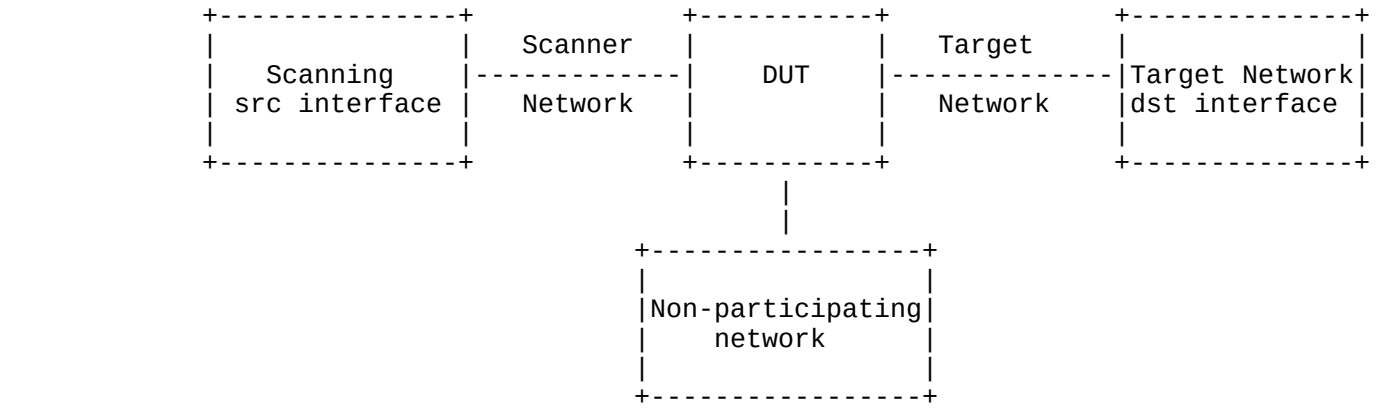
It is assumed that the latency for all network segments is neglible. By default, the target network's subnet shall be 64-bits in length, although some tests may involve increasing the prefix length.

Although packet size shouldn't have a direct impact, packet per second (pps) rates will have an impact. Smaller packet sizes should be utilized to facilitate higher packet per second rates.

For purposes of this test, the packet type being sent by the scanning device isn't important, although most scanning applications might want to send packets that would elicit responses from nodes within a subnet (such as an ICMPv6 echo request). Since it is not intended that responses be evoked from the target network node, such packets aren't necessary.

At the beginning of each test the intermediate node should be initialized. Minimally, the neighbor cache should be cleared.

Basic format of test network. Note that optional "non-participating network" is a third network not related to the scanner or target network.



4.1. Testing Interfaces

Two tester interfaces are configured for most tests:

- Scanning source (src) interface: This is the interface from which test packets are sourced. This interface sources traffic to destination IPv6 addresses on the target network from a single link-local address, similar to how an adjacent intermediate node would transit traffic through the intermediate node.

- Target network destination (dst) interface: This interface responds to neighbor solicitations as appropriate and confirms when an intermediate node has forwarded a packet to the interface for consumption. Where appropriate, the target network destination interface will respond to neighbor solicitations with a unique link-layer address per IPv6 address solicited.

5. Modifiers (variables)

5.1. Frequency of NDP triggering packets

The frequency of NDP triggering packets could be as high as the maximum packet per second rate that the scanner network will support (or is rated for). However, it may not be necessary to send packets at a particularly high rate and in fact a goal of testing could be to

identify if the DUT is able to withstand scans at rates which otherwise would not impact the performance of the DUT.

Optimistically, the scanning rate should be incremented until the DUT's performance begins deteriorating. Depending on the software and system being used to implement the scanning, it may be challenging to achieve a sufficient rate. Where this maximum threshold cannot be determined, the test results should note the highest rate tested and that DUT performance deteration was not noticed at this rate.

The lowest rate tested should be the rate for which packets can be expected to have an impact on the DUT -\u002D this value is of course, subjective.

6. Tests

6.1. Maximum number of valid hosts

This test evaluates how many hosts can be actively sending and receiving traffic on a network and still have connectivity across the intermediate node, calculated as the maximum number of valid hosts per second averaged over a 30 second period.

6.1.1. Test Streams

Two streams are defined:

1. Stream tester-new, sourced from the scanning source interface, sets up new addresses in the neighbor cache by sending packets, where each packet is sent to a unique IPv6 address by ascending order in the target network. If the packet is received at the target network interface, the address has been set up with an entry in the neighbor cache.

2. Stream tester-renew, sourced from the scanning source interface, sends traffic to existing addresses, where frequency of packets is between a millisecond and a second.

6.1.2. General Testing Procedure

1. Transit packets matching stream tester-new. Inittially, the rate for packets sent by tester-new should be a rate for which it is expected the intermediate node can transit. The rate should be increased until addresses are no longer being added to the neighbor cache as confirmed by neighbor solicitations no longer being sent by the intermediate node or the maximum bandwidth of the scanner or target network has been met, as measured by

comparing the traffic being transited with the maximum bandwidth of the links connecting to the intermediate node.

2. Once the maximum rate for stream tester-new has been determined, transit packets for stream tester-new until 30 seconds have evolved. Then send packets matching the tester-renew stream every second. This specific step should be continued until packets in either stream don't reach the target network destination interface.

3. If all packets from the tester-renew stream don't reach the target network destination interface before the completion of the 2 minute test, reduce the rate of the tester-new stream and repeat the test until all packets in both streams are received by the target network interface.

6.1.3. Discussion

The maximum number of valid hosts per second as calculated over a 30 second period is the rate for which all packets are transited to the target network interface in step 3 above.

This test is useful for confirming that there are no significant limitations in the intermediate node's capabilities, as defined by the intermediate node's intended deployment model and network. For example, if the intermediate node is intended for deployment where maximum traffic throughput is expected to be in the 1-Mbps range, it may not appropriate to require the intermediate node to perform acceptably where the traffic rate exceeds 10-Mbps.

6.2. Stable-state time

Given that it is possible to determine the maximum number of valid hosts per 30 second period without exceeding the capabilities of the tester or test network, this test determines how long it takes for the neighbor cache get into a reasonable state after the intermediate node has gotten into a state where packets are dropped. The period between when the disabling traffic is stopped and when the intermediate node no longer drops packets should be recorded.

6.2.1. Test streams

The test streams should be the same as those defined for "Maximum number of valid hosts."

6.2.2. General Testing Procedure

1. Replicate behavior in "maximum number of valid hosts" until packets are not being received by the target network destination interface.

2. Back off the rate to a point just below where packets previously were not received by the target network interface.

3. Measure the duration in seconds required until all packets are consistently received by the target network interface.

6.2.3. Discussion

This test confirms the rate at which an intermediate node recovers from a scanning incident where it’s neighbor cache and potentially other processes are overwhelmed.

6.3. NDP Prioritization: Behavior with stale neighbor entries

This test attempts to quantify how NDP prioritization, as discussed in RFC 6583 [RFC6583], is handled by the intermediate node. Priority should be given to hosts that have been seen before.

6.3.1. Test Streams

The test streams are the same as those defined for "Maximum number of valid hosts," with the addition of a "tester-unreachable" stream. This additional stream consists of sending packets for which the target network destination interface will not respond with neighbor advertisements.

6.3.2. General Testing Procedure

1. Send stream tester-new packets at a maximum rate as determined by "Maximum number of valid hosts."

2. Slow down stream tester-renew until one gets into refreshing every 6 seconds. If an address is in the "stale" state, it should get priority over new request.

3. Increase timer on stream tester-renew.

4. Stream tester-renew should always get responses. Stream tester-renew packets should always be recieved by the target network destination interface.

5. Stream tester-new should not always get responses. Stream tester-unreachable packets should not always be received by the target network destination interface.

6.3.3. Discussion

6.4. NDP Prioritizaton: Entries never present in neighbor cache

This test is identical to the first NDP prioritization test, except that reachability to nodes that never existed in the neighbor cache are confirmed.

6.4.1. Test Streams

6.4.2. General Testing Procedure

6.4.3. Discussion

NDP should prefer nodes that had previously been in the neighbor cache.

6.5. NDP Prioritization: Unreachable addresses only

This test evaluates the impact that scanning for non-existent addresses across an intermediate node has on the intermediate node’s ability to respond to NDP requests for valid nodes which had never been reached before.

6.5.1. Test Streams

There are two streams in this test: one consists of a significant flow of scanning traffic for non-existent nodes and the other comprises of attempting to reach existing nodes that had previously not had entries in the neighbor cache.

6.5.2. General Testing Procedure

1. Send stream tester-unreachable at a high rate for approximately 30 seconds, continuing traffic until the end of the test.

2. Send stream tester-new at a very low rate (perhaps once per second). Measure the rate at which the target network destination interface receives the packet.

6.5.3. Discussion

This test is intended to measure the real scenario where scanning is occuring on an otherwise idle network and there are a "handful" of real nodes on an end network which are being denied service because the NDP process cannot be completed in a timely manner.

7. Measurements Explicitly Excluded

These are measurements which aren't recommended because of the itemized reasons below:

7.1. DUT CPU Utilization

This measurement relies on the DUT to provide utilization information, which is subjective.

7.2. Malformed Packets

This benchmarking test is not intended to test DUT behavior in the presence of malformed packets.

8. DUT initialization

At the beginning of each test, the neighbor cache of the DUT should be initialized.

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

Benchmarking activities as described in this memo are limited to technology characterization using controlled stimuli in a laboratory environment, with dedicated address space and the constraints specified in the sections above.

The benchmarking network topology will be an independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network, or misroute traffic to the test management network.

Further, benchmarking is performed on a "black-box" basis, relying solely on measurements observable external to the DUT/SUT. Special capabilities SHOULD NOT exist in the DUT/SUT specifically for benchmarking purposes.

Any implications for network security arising from the DUT/SUT SHOULD be identical in the lab and in production networks.

11. Acknowledgements

12. Normative References

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, May 2008.

[RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.

Author’s Address

Bill Cerveny
Arbor Networks