

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 8, 2015

W. Cervený
Arbor Networks
R. Bonica
Juniper Networks
January 4, 2015

Benchmarking Neighbor Discovery
draft-cervený-bmwg-ipv6-nd-06

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 July 8, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Overview of Relevant NDP and Intermediate Node Behavior . . .	3
4. Test Setup	5
4.1. Testing Interfaces	6
5. Modifiers (Variables)	6
5.1. Frequency of NDP Triggering Packets	6
6. Tests	7
6.1. Stale Entry Time Determination	7
6.1.1. General Testing Procedure	7
6.2. Neighbor Cache Exhaustion Determination	7
6.2.1. General Testing Procedure	8
6.3. Dropped Flows Per Second	8
6.3.1. General Testing Procedure	8
7. Measurements Explicitly Excluded	8
7.1. DUT CPU Utilization	9
7.2. Malformed Packets	9
8. DUT Initialization	9
9. IANA Considerations	9
10. Security Considerations	9
11. Acknowledgements	9
12. References	10
12.1. Normative References	10
12.2. Informative References	10
Authors' Addresses	10

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]

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

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

Stale Entry Time This is the duration for which a neighbor cache entry marked "Reachable" will continue to be marked "Reachable" if an update for the address is not received.

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 occurring 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 solicitations. 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.

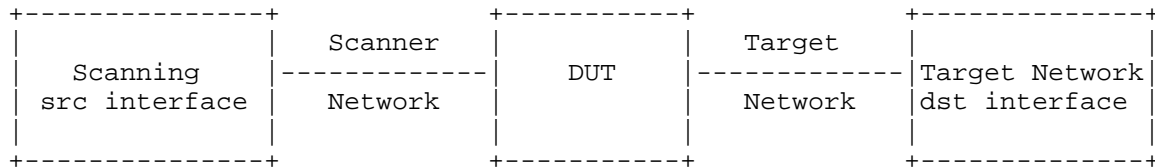
It is assumed that the latency for all network segments is negligible. By default, the target network's subnet shall be 64-bits in length, although some tests may involve increasing the prefix length.

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:

- o 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.
- o 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 can 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. In fact, a non-benchmarking 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 deterioration 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 -- this value is of course, subjective.

6. Tests

6.1. Stale Entry Time Determination

This test determines the time interval when the intermediate node (DUT) identifies an address as stale.

RFC 4861, section 6.3.2 [RFC4861] states that an address can be marked "stale" at a random value between 15 and 45 seconds (as defined via constants in the RFC). This test confirms what value is being used by the intermediate node. Note that RFC 4861 states that this random time can be changed "at least every few hours."

6.1.1. General Testing Procedure

1. Send a packet from the scanning source interface to an address in target network. Observe that the intermediate node sends a neighbor solicitation to the solicited-node multicast address on the target network, for which tester destination interface should respond with a neighbor advertisement. The intermediate node should create an entry in neighbor cache for the address, marking the address as "reachable". As this point, the packet should be forwarded to the tester destination interface.
2. After the neighbor advertisement from the destination tester interface in step one, no more neighbor advertisements from the tester destination interface should be allowed.
3. Continue sending packets from the scanning source interface to the same address in the target network.
4. Note the time at which the DUT no longer sends packets. The stale timer value will be the period of time between when the DUT received the first neighbor advertisement above and the point at which the DUT no longer forwards packets for this flow to the tester destination interface.

6.2. Neighbor Cache Exhaustion Determination

Discover the point at which the neighbor cache is exhausted and evaluate intermediate node behavior when this threshold is reached. If possible, the stale timer value should be locked down to a large value. A side-effect of this test is to confirm that intermediate node behaves correctly; in particular, it shouldn't crash.

Note that some intermediate nodes may restrict the frequency of allowed neighbor discovery packets transmitted. The maximum allowed packets per second must either be set to a value which doesn't impact the outcome of the test must allow for this restriction.

6.2.1. General Testing Procedure

1. At a very fast rate, send packets incrementally to valid unique addresses in the target network, within stale entry time period. Simultaneously, send packets for addresses previously added to the neighbor cache. The neighbor cache has been exhausted when previously added addresses must be re-discovered with a neighbor solicitation (within the stale entry time period).
2. Observe what happens when one address greater than the maximum neighbor cache size ("n") is reached. When "n+1" is reached, if either the first or most recent cache entry are dropped, this may be acceptable.
3. Confirm intermediate node doesn't crash when "n+1" is reached.

6.3. Dropped Flows Per Second

This test determines the rate at which flows are dropped once the neighbor cache size is exceeded. The metric for this test is the number of flows which are dropped in a minute.

6.3.1. General Testing Procedure

1. Send packets incrementally to unique valid addresses in the target network, within stale entry time period. The number of unique valid addresses may be as high as the size of the neighbor cache, but may be the number of nodes that would be expected in a deployed network. Continue sending packets to previously cached addresses.
2. Send packets incrementally to unique invalid addresses (addresses without valid node in target network), until the intermediate node crashes, packets are no longer accepted or existing flows to unique valid addresses are dropped.

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

Helpful comments and suggestions were offered by Al Morton, Joel Jaeggli, Nalini Elkins, Scott Bradner, and Ram Krishnan, on the BMWG e-mail list and at BMWG meetings. Precise grammatical corrections and suggestions were offered by Ann Cerveney.

12. References

12.1. 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.

12.2. Informative References

- [RFC7048] Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, January 2014.

Authors' Addresses

Bill Cerveney
Arbor Networks
2727 South State Street
Ann Arbor, MI 48104
USA

Email: wcerveney@arbor.net

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20170
USA

Email: rbonica@juniper.net

Network Working Group
Internet Draft
Intended status: Informational
Expires: June 2013
December 1, 2012

B. Constantine
JDSU
T. Copley
Level-3
R. Krishnan
Brocade Communications

Traffic Management Benchmarking
draft-constantine-bmwg-traffic-management-00.txt

Status of this Memo

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on July 5, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This framework describes a practical methodology for benchmarking the traffic management capabilities of networking devices (i.e. policing, shaping, etc.). The goal is to provide a repeatable test method that objectively compares performance of the device's traffic management capabilities and to specify the means to benchmark traffic management with representative application traffic.

Table of Contents

1. Introduction.....	3
1.1. Traffic Management Overview.....	3

2. Conventions used in this document.....	5
3. Scope and Goals.....	6
4. Traffic Benchmarking Metrics.....	7
4.1. Metrics for Stateless Traffic Tests.....	7
4.2. Metrics for Stateful Traffic Tests.....	8
5. Tester Capabilities.....	9
5.1. Stateless Test Traffic Generation.....	9
5.2. Stateful Test Pattern Generation.....	9
5.2.1. TCP Test Pattern Definitions.....	10
6. Traffic Benchmarking Methodology.....	12
6.1. Policing Tests.....	12
6.2. Queue Tests.....	13
6.2.1. Testing Queue with Stateless Traffic.....	13
6.2.2. Testing Queue with Stateful Traffic.....	14
6.3. Shaper tests.....	14
6.3.1. Testing Shaper with Stateless Traffic.....	15
6.3.2. Testing Shaper with Stateful Traffic.....	16
6.4. Congestion Management tests.....	17
6.4.1. Testing Congestion Management with Stateless Traffic.....	17
6.4.2. Testing Congestion Management with Stateful Traffic.....	17
7. Security Considerations.....	20
8. IANA Considerations.....	20
9. Conclusions.....	20
10. References.....	20
10.1. Normative References.....	20
10.2. Informative References.....	21
11. Acknowledgments.....	21
12. First Appendix.....	21

1. Introduction

Traffic management (i.e. policing, shaping, etc.) is an increasingly important component in today's networks. There is no framework to benchmark these features although some standards address specific areas. This draft provides a framework to conduct repeatable traffic management benchmarks for devices and systems in a lab environment. The benchmarking framework can also be used as a test procedure to assist in the tuning of Quality of Service (QoS) parameters before field deployment. In addition to Layer 2/3 benchmarking, techniques to define Layer 4 traffic test patterns are presented that can benchmark the traffic management technique(s) under realistic conditions.

1.1. Traffic Management Overview

In general, a device with traffic management capabilities performs the following QoS functions:

- . Traffic classification: identifies traffic according to various QoS rules (i.e. VLAN, DSCP, etc.) and marks this traffic internally to the network device (for traffic management processing)
- . Traffic policing: rate limits traffic that enters a router according to the traffic classification. If the traffic exceeds the contracted Service Level Agreement (SLA), the traffic is either dropped or remarked and sent onto to the next network node
- . Traffic shaping: is a traffic control measure of actively buffering and metering the output rate of traffic in an attempt to adapt bursty traffic to the SLA.
- . Traffic Scheduling: provides QoS within the network device by storing packets in various types of queues and applies a dispatching algorithm to assign the forwarding sequence of packets.
- . Congestion Management: monitors the status of internal queues and actively drops packets, which causes the sending hosts to back-off and in turn can alleviate queue congestion.

The following diagram is a generic model of the traffic management capabilities within a network device. It is not intended to represent all variations of manufacturer traffic management capabilities, but provide context to this test framework.

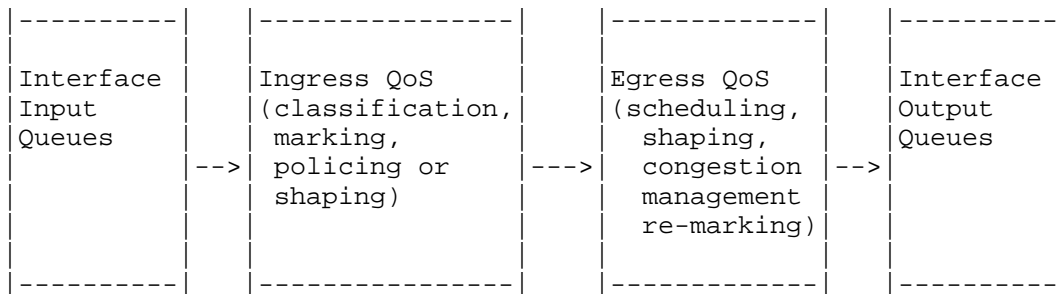


Figure 1: Generic Model of Traffic Management capabilities within a network device

(TC comment: A couple other things that a traffic management device must be able to perform Is Marking / Remarking / encapsulation. I also think we should be looking at the performance that these types of functions add to the packet.)

2. Conventions used in this document

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].

The following acronyms are used:

BDP: Bandwidth Delay Product

CBS: Committed Burst Size

CIR: Committed Information Rate

DUT: Device Under Test

EBS: Exceeded Burst Size

EIR: Exceeded Information Rate

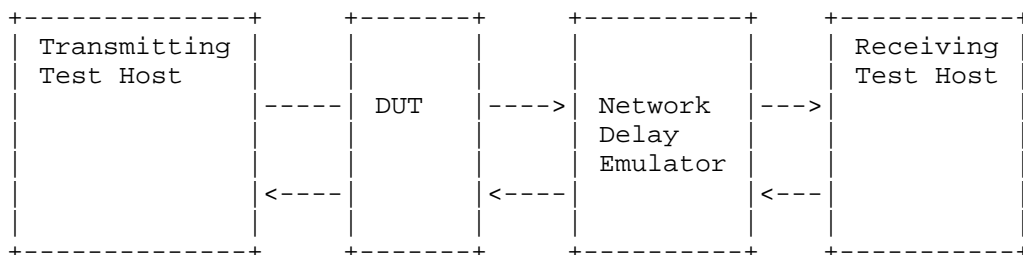
QoS: Quality of Service

RED: Random Early Discard

RTT: Round Trip Time

WRED: Weighted Random Early Discard

The following is the description of the lab set-up for the traffic management tests:



As shown the test diagram, the framework supports uni-directional and bi-directional traffic management tests.

Also note that the Network Delay Emulator (NDE) should be passive in nature such as a fiber spool. This is recommended to eliminate the potential effects that an active delay element (i.e. test impairment generator) may have on the test flows. In the case that a fiber spool is not practical due to the desired latency, an active NDE must be independently verified to be capable of adding the configured delay without loss. This requirement will vary from test to test on desired traffic speed and should be calibrated before any test requiring delay, which can add a significant additional amount of testing to each step.

3. Scope and Goals

The scope of this work is to develop a framework for benchmarking and testing the traffic management capabilities of network devices in the lab environment. These network devices may include but are not limited to:

- Switches (including Layer 2/3 devices)
- Routers
- Firewalls

Essentially, any network device that performs traffic management as defined in section 1.1 can be benchmarked or tested with this framework.

Within this framework, the metrics are defined for each traffic management test but do not include pass / fail criterion, which is not within the charter of BMWG. This framework does not attempt to rate the performance of one manufacturer's network equipment versus another, but only to provide benchmarks to conduct repeatable, comparative testing.

A goal of this framework is to define specific stateless traffic ("packet blasting") tests to conduct the benchmark tests and also to derive stateful test patterns (TCP or application layer) that can also be used to further benchmark the performance of applicable traffic management techniques such as traffic shaping and congestion management techniques such as RED/WRED. In cases where the network

device is stateful in nature (i.e. firewall, etc.), stateful test pattern traffic is the only option.

And finally, this framework will provide references to open source tools that can be used to provide the stateless traffic generation capabilities and the stateful emulation capabilities referenced above.

4. Traffic Benchmarking Metrics

The metrics to be measured during the benchmarks are divided into two (2) sections: packet layer metrics used for the stateless traffic testing and metrics used for the stateful traffic testing

4.1. Metrics for Stateless Traffic Tests

The following are the metrics to be used during the stateless traffic benchmarking components of the tests:

- Burst Size Achieved (BSA): for the traffic policing and network queue tests, the tester will be configured to send bursts to test either the Committed Burst Size (CBS) or Exceeded Burst Size (EBS) of a policer or the queue / buffer size configured in the DUT. The Burst Size Achieved metric is a measure of the actual burst size received at the egress port of the DUT with no lost frames. As an example, the CBS of a DUT is 64KB and after the burst test, only a 63 KB can be achieved without frame loss. Then 63KB is the BSA.
- Lost Frames (LF): For all traffic management tests, the tester will transmit the test frames into the DUT ingress port and the number of frames received at the egress port will be measured. The difference between frames transmitted into the ingress port and received at the egress port is the number of lost frames as measured at the egress port. These frames must have unique identifiers such that only the test frames are measured.
- Out of Sequence Frames (OOS): in additions to LF metric, the test frames must be monitored for sequence and the out-of-sequence (OOS) frames will be counted per RFC-???? or is this ITU??.
- Frame Delay (FD): the Frame Delay metric is the difference between the timestamp of the received egress port frames and the frames transmitted into the ingress port and specified in ITU-1564.
- Frame Delay Variation (FDV): the Frame Delay Variation metric is the variation between the timestamp of the received egress port frames and specified in ITU-1564.

(Note, we need to consider bi-directional nature of the tests and metrics)

4.2. Metrics for Stateful Traffic Tests

The stateful metrics will be based on RFC 6349 TCP metrics and will include the following:

- TCP Test Pattern Execution Time: RFC 6349 defined the TCP Transfer Time for bulk transfers, which is simply the measured time to transfer bytes across single or concurrent TCP connections. The TCP test patterns used in traffic management tests will be bulk transfer and interactive in nature; these test patterns simulate delay-tolerant applications like FTP, streaming video etc.. The TTPET will be the measure of the time for a single execution of a TTPET. Average, minimum, and maximum times will be measured.

- TCP Efficiency: after the execution of the TCP Test Pattern, TCP Efficiency represents the percentage of Bytes that were not retransmitted.

Transmitted Bytes - Retransmitted Bytes

TCP Efficiency % = ----- X 100

Transmitted Bytes

Transmitted Bytes are the total number of TCP Bytes to be transmitted including the original and the retransmitted Bytes.

- Buffer Delay: represents the increase in RTT during a TCP test versus the baseline DUT RTT (non congested, inherent latency). The average RTT is derived from the total of all measured RTTs during the actual test at every second divided by the test duration in seconds.

Total RTTs during transfer

Average RTT during transfer = -----

Transfer duration in seconds

Average RTT during Transfer - Baseline RTT

Buffer Delay % = ----- X 100

Baseline RTT

5. Tester Capabilities

The testing capabilities of the traffic management test environment are divided into two (2) sections: stateless traffic testing and stateful traffic testing

5.1. Stateless Test Traffic Generation

The test set must be capable of generating test traffic at up to the link speed of the DUT. The test set must be calibrated to verify that it will not drop any frames. The test set's inherent FD and FDV must also be calibrated and subtracted from the FD and FDV metrics.

The test set must support the encapsulation to be tested such as VLAN, Q-in-Q, MPLS, etc.

The open source tool "iperf" can be used to generate stateless UDP traffic and is discussed in Appendix A. Since iperf is a software based tool, there will be performance limitations at higher link speeds. Careful calibration of any test environment using iperf is important. At higher link speeds, it is recommended to select commercial hardware based packet test equipment.

5.2. Stateful Test Pattern Generation

The TCP test host will have many of the same attributes as the TCP test host defined in RFC 6349. The TCP test host may be a standard computer or a dedicated communications test instrument. In both cases, it must be capable of emulating both a client and a server.

For any test using stateful TCP test traffic, the Network Delay Emulator (NDE) function from the lab set-up must be used in order to provide a meaningful BDP. As referenced in section 2, the target traffic rate and configured RTT must be verified independently using just the NDE for all stateful tests (to ensure the NDE can delay without loss).

The TCP test host must be capable to generate and receive stateful TCP test traffic at the full link speed of the DUT. As a general rule of thumb, testing TCP Throughput at rates greater than 100 Mbps may require high performance server hardware or dedicated hardware based test tools.

(TC comment: You mention that a device to do rates greater than 100Mbit may require a high performance server. We also need to discuss how window Sizes or flows impact that.)

The TCP test host must allow adjusting both Send and Receive Socket Buffer sizes. The Socket Buffers must be large enough to fill the BDP for bulk transfer TCP test application traffic.

Measuring RTT and retransmissions per connection will generally require a dedicated communications test instrument. In the absence of dedicated hardware based test tools, these measurements may need to be conducted with packet capture tools, i.e. conduct TCP Throughput tests and analyze RTT and retransmissions in packet captures.

The TCP implementation used by the test host must be specified in the test results (i.e. OS version, i.e. LINUX OS kernel using TCP New Reno, TCP options supported, etc).

While RFC 6349 defined the means to conduct throughput tests of TCP bulk transfers, the traffic management framework will extend TCP test execution into interactive TCP application traffic. Examples include email, HTTP, business applications, etc. This interactive traffic is not uni-directional in nature but is chatty.

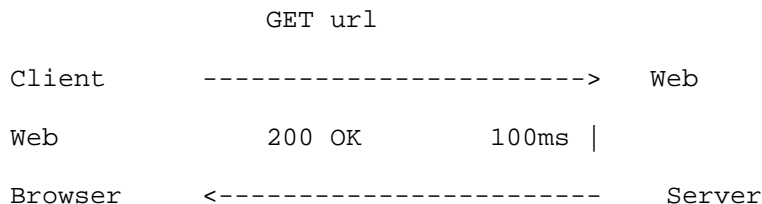
The test host must not only support bulk TCP transfer application traffic but this chatty traffic since the both stress traffic management techniques in very different ways. This is due to the non-uniform, bursty nature of chatty applications versus the relatively uniform nature of bulk transfers (the bulk transfer smoothly stabilizes to equilibrium state under lossless conditions).

While iperf is an excellent choice for TCP bulk transfer testing, the open source tool "Flowgrind" is applicable to interactive TCP flows and is also referenced in Appendix A. Flowgrind is client server based and emulates interactive applications at the TCP layer. As with any software based tool, the performance must be qualified to the link speed to be tested. Commercial test equipment should be considered for reliable results at higher links speeds.

5.2.1. TCP Test Pattern Definitions

As mentioned in the goals of this framework, techniques to define Layer 4 traffic test patterns will be defined to benchmark the traffic management technique(s) under realistic conditions. Some network devices such as firewalls, will not process stateless test traffic which is another reason that stateful TCP test traffic must be used.

An application can be fully emulated to Layer 7 but this framework proposes that stateful TCP test patterns be used to provide granular and repeatable control for the benchmarks. The following diagram illustrates a simple Web Browsing application (HTTP).



In this example, the Client Web Browser (Client) requests a URL and then the Web Server delivers the web page contents to the Client (after a Server delay of 100 msec). This synchronous, "request / response" behavior is intrinsic to most TCP based applications such as Email (SMTP), File Transfers (FTP and SMB), Database (SQL), Web Applications (SOAP), etc. The impact to the network elements is due to the multitudes of Clients and the variety of bursty traffic, which stresses network resources such as buffers, shapers, and other QoS management techniques. The actual emulation of the specific application protocols is not required and TCP test patterns can be defined to mimic the application behavior.

This framework does not specify a fixed set of TCP test patterns, but does provide examples in Appendix B. There are two (2) techniques recommended by this framework to develop standard TCP test patterns for traffic management benchmarking.

The first technique involves modeling techniques, which have been described in "3GPP2 C.R1002-0 v1.0" and describe the behavior of HTTP, FTP, and WAP applications at the TCP layer. The models have been defined with various mathematical distributions for the Request/Response bytes and inter-request gap times. The Flowgrind tool (Appendix A) supports many of the distributions and is a good choice as long as the processing limits of the server platform are taken into consideration.

The second technique is to conduct packet captures of the applications to test and then to statefully play the application back at the TCP layer. The TCP playback includes the request byte size, response byte size, and inter-message gaps at both the client and the server. The advantage of this method is that very realistic test patterns can be defined based off of real world application traffic.

Appendix B provides an overview of the modeling technique with Flowgrind, capture technique with TCP playback, and some representative application traffic that can be used with either technique.

(TC comment: In addition to application test patterns, I'd also like to see some of the standard ways mentioned like 2544 all 1's all F's all 0's and the Alternating)

6. Traffic Benchmarking Methodology

The traffic benchmarking methodology uses the test set-up from section 2 and metrics defined in section 4. Each test should be run for a minimum test time of 5 minutes.

6.1. Policing Tests

The intent of the policing tests is to verify the policer performance parameters of CIR-CBS and EIR-EBS. The tests will verify that the device can handle the CIR rate with CBS and the EIR rate with EBS and will use back-back frame testing concepts from RFC 2544 (but adapted to burst size algorithms and terminology). Also MEF-14,19,37 provide some basis for specific components of this test.

Policing tests will only use stateless traffic since a policer only operates at Layer 2. Stateful TCP test traffic would not yield any benefit to test a policer.

The policer test traffic shall follow the traffic profile as defined in MEF 10.2. Specifically, the stateless traffic shall be transmitted at the link speed within the time interval of the policer. In MEF 10.2, this time interval is defined as:

$$T_c = (CBS * 8) / CIR \text{ or}$$

$$T_e = (EBS * 8) / EIR$$

As an example, consider a CBS of 64KB and CIR of 100 Mbps on a 1GigE physical link. The T_c equates to 5.12 msec and the 64KB burst should be transmitted into the ingress port at full GigE rate, then wait for 5.12 msec for the next burst, etc.

The metrics defined in section 4.1 shall be measured at the egress port and recorded; the primary result is to verify the BSA and that no frames are dropped.

In addition to verifying that the policer allows the specified CBS and EBS bursts to pass, the policer test must verify that the policer will police at the specified CBS/EBS values.

For this portion of the test, the CBS/EBS value should be incremented by 1000 bytes higher than the configured CBS and that the egress port measurements must show that the majority of frames are dropped.

6.2. Queue Tests

The queue tests are similar in nature and can be covered with the same test technique for the stateless traffic tests. There are not CIR-CBS, EIR-EBS parameters for network device queues so only the CBS component of the policer tests should be applied to pure queue tests.

Since device queues / buffers are generally an egress function, this test framework will discuss testing at the egress (although the technique can be applied to ingress side queues).

6.2.1. Testing Queue with Stateless Traffic

A network device queue is memory based unlike a policing function, which is token or credit based. However, the same concepts from section 6.1 can be applied to testing network device queue.

The device's network queue should be configured to the desired size in KB (queue length, QL) and then stateless traffic should be transmitted to test this QL.

The transmission interval (Ti) can be defined for the traffic bursts and is based off of the QL and Bottleneck Bandwidth (BB) of the egress interface. The equation is similar to the Tc / Te time interval discussed in the policer section 6.1 and is as follows:

$$Ti = QL * 8 / BB$$

Important to note that the assumption is that the aggregate ingress throughput is higher than the BB or the queue test is not relevant since there will not be any over subscription.

The stateless traffic shall be transmitted at the link speed within the Ti time interval. The metrics defined in section 4.1 shall be measured at the egress port and recorded; the primary result is to verify the BSA and that no frames are dropped.

6.2.2. Testing Queue with Stateful Traffic

To provide a more realistic benchmark and to test queues in layer 4 devices such as firewalls, stateful traffic testing is recommended for the queue tests. Stateful traffic tests will also utilize the Network Delay Emulator (NDE) from the network set-up configuration in section 2.

The BDP of the TCP test traffic must be calibrated to the QL of the device queue. The BDP is equal to:

$BB * RTT / 8$ (in bytes)

The NDE must be configured to an RTT value which is great enough to allow the BDP to be greater than QL. An example test scenario is defined below:

- Ingress link = Gige
- Egress link = 100 Mbps (BB)
- QL = 32KB

$RTT(min) = QL * 8 / BB$ and would equal 2.56 msec and the BDP = 32KB

In this example, one (1) TCP connection with window size / SSB of 32KB would be required to test the QL of 32KB. This Bulk Transfer Test can be accomplished using iperf as described in Appendix A.

The test metrics will be recorded per the stateful metrics defined in 4.2, primarily the TCP Test Pattern Execution Time (TTPET), TCP Efficiency, and Buffer Delay.

In addition to a Bulk Transfer Test, it is recommended to run the Bursty Test Pattern from appendix B at a minimum. Other tests from include: Small Web Site, Email, Citrix, etc.

The traffic is bi-directional - the same queue size is assumed for both directions.

6.3. Shaper tests

The intent of the shaper tests is to verify the shaper performance parameters of shape rate (SR) and shape burst size (SBS). The tests will verify that the device can handle the CIR rate with CBS and smooth the traffic bursts to the shaper rate.

Since device queues / buffers are generally an egress function, this test framework will discuss testing at the egress (although the technique can be applied to ingress and internal queues).

A network device's traffic shaper will generally either shape to an average rate or provide settings similar to a policer to set the CIR and CBS. In the context of a shaper, the CBS indicates the size of the burst that the shaper can accept within the shaping time interval.

The shaping time interval depends upon whether the average method or CIR/CBS method is supported by the network device. If only the average method is supported, then the shaping time interval (period at which bursts will be shaped) must be determined through manufacturer product specifications.

For shapers that utilize the CIR/CBS method, the shaper time interval is the same as Tc for the policer which is indicated in section 6.1.

(TC comment: We need to be able to measure FD over a shaper. That should be the ms of queue depth.)

6.3.1. Testing Shaper with Stateless Traffic

A traffic shaper is memory based like a queue, but with the added intelligence of an active shaping element. The same concepts from section 6.2 (Queue testing) can be applied to testing network device shaper.

The device's traffic shaping function should be configured to the desired SR and SBS (for devices supporting this parameter) and then stateless traffic should be transmitted to test the SBS.

The same example from section 6.1 is used with SBS of 64KB and CIR of 100 Mbps; both ingress and egress ports are GigE. The Tc equates to 5.12 msec and the 64KB burst should be transmitted into the ingress port at full GigE rate, then wait for 5.12 msec for the next burst, etc.

While the ingress traffic will burst up to GigE link speed for the duration of the SBS burst, the egress traffic should be smoothed or averaged to the CIR rate on the egress port.

In addition to the egress metrics to be measured per section 4.1, the stateless shaper test shall record:

- Average shaper rate on the egress port

- Variation (min, max) around the shaper rate

6.3.2. Testing Shaper with Stateful Traffic

To provide a more realistic benchmark and to test queues in layer 4 devices such as firewalls, stateful traffic testing is also recommended for the shaper tests. Stateful traffic tests will also utilize the Network Delay Emulator (NDE) from the network set-up configuration in section 2.

The BDP of the TCP test traffic must be calculated as described in section 6.2.2. To properly stress network buffers and the traffic shaping function, the cumulative TCP window should exceed the BDP which will stress the shaper. BDP factors of 1.1 to 1.5 are recommended, but the values are the discretion of the tester and should be documented.

By cumulative TCP window, this equates to:

TCP window size* for each connection x number of connections

* TCP window size is used per RFC 6349 and is the minimum of the TCP WIN and the Send Socket Buffer (SSB)

Example, if the BDP is equal to 256 Kbytes and a connection size of 64Kbytes is used for each connection, then it would require four (4) connections to fill the BDP and 5-6 connections (over subscribe the BDP) to stress test the traffic shaping function.

Two types of tests are recommended: Bulk Transfer test and Bursty Test Pattern as documented in Appendix B at a minimum. Other tests from include: Small Web Site, Email, Citrix, etc.

The test metrics will be recorded per the stateful metrics defined in 4.2, primarily the TCP Test Pattern Execution Time (TTPET), TCP Efficiency, and Buffer Delay.

The traffic is bi-directional involving multiple egress ports.

In addition to the egress metrics to be measured per section 4.2, the stateful shaper test shall record:

- Average shaper rate on each egress port
- Variation (min, max) around the shaper rate

6.4. Congestion Management tests

The intent of the congestion management tests is to benchmark the performance of various active queue management (AQM) discard techniques such as RED, WRED, etc. AQM techniques vary, but the basic principal is to discard traffic before the queue overflows (FIFO). This discard in effect sends congestion notification warning to protocols such as TCP, which causes TCP to back-off and ideally improves aggregate throughput by preventing global TCP session loss (tail drop).

The key parameter for AQM techniques is the discard threshold of the queue. (RK comment: The discard is also probabilistic http://en.wikipedia.org/wiki/Random_early_detection). In some network devices, this discard threshold is discretely configurable (i.e. percent of queue depth) and in others the discard threshold is intrinsic to the AQM technique itself.

As such AQM benchmark testing may involve a certain level of characterization experiments in which the burst size transmitted may increase as a portion of the queue depth.

6.4.1. Testing Congestion Management with Stateless Traffic

If the queue discard threshold is discretely configurable, then the stateless burst techniques described in sections 6.2.1 (queuing tests) can be applied directly to the AQM tests. In other words, the queue will be over-subscribed and burst transmitted into the device within the T_i interval as defined in 6.2.1

For AQM techniques where the discard threshold is not discretely configurable, then a stair case ramp is recommended to characterize and compare the AQM technique between devices. For example if the $QL = 32KB$, then it would be reasonable to test with burst sizes in increments of 25% to include 8KB, 16KB, 32KB and record the metrics per section 4.2. (RK comment: We should send a burst and examine if there are discontinuous drops - in the case of tail drop, the drops will be continuous)

6.4.2. Testing Congestion Management with Stateful Traffic

Similar to the Queue tests (section 6.2) and Shaper tests (section 6.3), stateful traffic tests will utilize the Network Delay Emulator (NDE) to add RTT. The RTT should be configured such that BDP would equal at least 64KB.

The key metric to be measured for the stateful tests is the TCP Test Pattern Execution Time (TTPET). AQM is intended to improve TCP performance by preventing tail-drop and it is the TTPET that provides the appropriate metric to compare the AQM techniques between vendors.

An example is as follows: transmit n TCP flows using the AQM Test Pattern (reference Appendix B) and measure the TTPET with and without AQM enabled. The number of flows should be configured to exceed the BDP with recommended oversubscription within the 1.1 - 1.5 range.

The test metrics will be recorded per the stateful metrics defined in 4.2, primarily the TCP Test Pattern Execution Time (TTPET), TCP Efficiency, and Buffer Delay.

(TCP miscellaneous comments:

You don't talk about impacts of RED on independent flows on testing congestion management Do certain flows get impacted more than others.

There is no discussion of SPQ versus WFQ, or any mention of QOS measurements. We also need To make recommendations on QOS parameters / variables for acting on.

There was no discussion of UDP

There was no discussion calculating window size

)

Appendix A: Open Source Tools for Traffic Management Testing

This traffic management framework specified that both stateless and stateful traffic testing be conducted. Two (2) open source tools that can be used are iperf and Flowgrind to accomplish many of the tests proposed in this framework.

Iperf can generate UDP or TCP based traffic; a client and server must both run the iperf software in the same traffic mode. The server is set up to listen and then the test traffic is controlled from the client. Both uni-directional and bi-directional concurrent testing are supported.

The UDP mode can be used for the stateless traffic testing. The target bandwidth, frame size, UDP port, and test duration can be controlled. A report of bytes transmitted, frames lost, and delay variation are provided by the iperf receiver.

The TCP mode can be used for stateful traffic testing to test bulk transfer traffic. The TCP Window size (which is actually the SSB), the number of connections, the frame size, TCP port and the test duration can be controlled. A report of bytes transmitted and throughput achieved are provided by the iperf sender.

Flowgrind is a distributed network performance measurement tool. Using the flowgrind controller, tests can be setup between hosts running flowgrind. For the purposes of this traffic management testing framework, the key benefit of Flowgrind is that it can emulate non-bulk transfer applications such as HTTP, Email, etc. This is due to fact that Flowgrind supports the concept of request and response behavior while iperf does not.

Traffic generation options include the request size, response size, inter-request gap, and response time gap. Additionally, various distribution types are supported including constant, normal, exponential, pareto, etc. These powerful traffic generation parameters facilitate the modeling of complex application test patterns at the TCP layer which are discussed in Appendix B.

Since these tools are software based, the host hardware must be qualified to be capable of generating the target traffic loads without frame loss and within the frame delay variation threshold.

Appendix B: Stateful TCP Test Patterns

This framework does not specify a fixed set of TCP test patterns, but proposes two (2) techniques to develop standard TCP test patterns for traffic management benchmarking and provides examples of the following test patterns:

- Bulk: generate concurrent TCP connections transmit an aggregate number of in-flight data bytes (i.e. could be the BDP). Guidelines from RFC 6349 are used to create this traffic model.
- Bursty: generate precise burst pattern within a single or multiple TCP sessions. The idea is for TCP to establish equilibrium on a connection(s) and then to burst application bytes at a defined burst size.
- AQM: generate various burst sizes within an TCP session, spacing the bursts apart such that size of the burst size achieved (BSA) can be easily determined. In a sense, this could be considered a TCP stair case or ramp test.

- Small Web Site: mimic the request and response (chatty) and bulk transfer (page download) behavior of a less complex web site. This example uses the modeling technique with Flowgrind to generate this TCP test pattern.

- Cirix: mimic very chatty behavior of Citrix. This example uses the packet capture technique to model the behavior and discusses the requirements for test tools to playback the packet capture statefully.

TBD: Detailed definitions for each of the test patterns listed above.

From these examples, users can extrapolate others that may be more suitable to their intended test needs.

7. Security Considerations

8. IANA Considerations

9. Conclusions

10. References

10.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

10.2. Informative References

11. Acknowledgments

12. First Appendix

Authors' Addresses

Barry Constantine

JDSU, Test and Measurement Division

Germantown, MD 20876-7100, USA

Phone: +1 240 404 2227

Email: barry.constantine@jdsu.com

Timothy Copley

Level 3 Communications

14605 S 50th Street

Phoenix, AZ 85044

Email: Timothy.copley@level3.com

Ram Krishnan

Brocade Communications

San Jose, 95134, USA

Phone: +001-408-406-7890

Email: ramk@brocade.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 5, 2013

A. Morton
AT&T Labs
June 3, 2013

IMIX Genome: Specification of variable packet sizes for additional
testing
draft-ietf-bmwg-imix-genome-05

Abstract

Benchmarking Methodologies have always relied on test conditions with constant packet sizes, with the goal of understanding what network device capability has been tested. Tests with constant packet size reveal device capabilities but differ significantly from the conditions encountered in operational deployment, and so additional tests are sometimes conducted with a mixture of packet sizes, or "IMIX". The mixture of sizes a networking device will encounter is highly variable and depends on many factors. An IMIX suited for one networking device and deployment will not be appropriate for another. However, the mix of sizes may be known and the tester may be asked to augment the fixed size tests. To address this need, and the perpetual goal of specifying repeatable test conditions, this draft defines a way to specify the exact repeating sequence of packet sizes from the usual set of fixed sizes, and other forms of mixed size specification.

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 December 5, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
2. Scope and Goals	4
3. Specification of the IMIX Genome	5
4. Specification of a Custom IMIX	7
5. Reporting Long or Pseudo-Random Packet Sequences	8
6. Security Considerations	9
7. IANA Considerations	9
8. Acknowledgements	9
9. References	9
9.1. Normative References	9
9.2. Informative References	10
Author's Address	10

1. Introduction

This memo defines a method to unambiguously specify the sequence of packet sizes used in a load test.

Benchmarking Methodologies [RFC2544] have always relied on test conditions with constant packet sizes, with the goal of understanding what network device capability has been tested. Tests with the smallest size stress the header processing capacity, and tests with the largest size stress the overall bit processing capacity. Tests with sizes in-between may determine the transition between these two capacities.

Streams of constant packet size differ significantly from the conditions encountered in operational deployment, and so additional tests are sometimes conducted with a mixture of packet sizes. The set of sizes used is often called an Internet Mix, or "IMIX" [Spirent], [IXIA], [Agilent].

The mixture of sizes a networking device will encounter is highly variable and depends on many factors. An IMIX suited for one networking device and deployment will not be appropriate for another. However, the mix of sizes may be known and the tester may be asked to augment the fixed size tests. The references above cite the original studies and their methodologies. Similar methods can be used to determine new size mixes present on a link or network. We note that the architecture for IP Flow Information Export [RFC5470] provides one method to gather packet size information on private networks.

To address this need, and the perpetual goal of specifying repeatable test conditions, this memo proposes a way to specify the exact repeating sequence of packet sizes from the usual set of fixed sizes: the IMIX Genome. Other, less exact forms of size specification are also recommended for extremely complicated or customized size mixes. We apply the term "genome" to infer that the entire test packet size sequence can be replicated if this information is known, a parallel to the information needed for biological replication.

This memo takes the position that it cannot be proven for all circumstances that the sequence of packet sizes does not affect the test result, thus a standardized specification of sequence is valuable.

2. Scope and Goals

This memo defines a method to unambiguously specify the sequence of packet sizes that have been used in a load test, assuming that a

relevant mix of sizes is known to the tester and the length of the repeating sequence is not very long (<100 packets).

The IMIX Genome will allow an exact sequence of packet sizes to be communicated as a single-line name, resolving the current ambiguity with results that simply refer to "IMIX". This aspect is critical because no ability has been demonstrated to extrapolate results from one IMIX to another IMIX, even when the mix varies only slightly from another IMIX, and certainly no ability to extrapolate results to other circumstances.

While documentation of the exact sequence is ideal, the memo also covers the case where the sequence of sizes is very long or may be generated by a pseudo-random process.

It is a colossal non-goal to standardize one or more versions of the IMIX. This topic has been discussed on many occasions on the `bmwg-list` [`IMIXonList`]. The goal is to enable customization with minimal constraints while fostering repeatable testing once the fixed size testing is complete. Thus, the requirements presented in this specification, expressed in [RFC2119] terms, are intended for those performing/reporting laboratory tests to improve clarity and repeatability.

3. Specification of the IMIX Genome

The IMIX Genome is specified in the following format:

IMIX - 123456...x

where each number is replaced by the letter corresponding to the size of the packet at that position in the sequence. The following table gives the letter encoding for the [RFC2544] standard sizes (64, 128, 256, 512, 1024, 1280, and 1518 bytes) and "jumbo" sizes (2112, 9000, 16000). Note that the 4 octet Ethernet frame check sequence may fail to detect bit errors in the larger jumbo frames, see [jumbo].

Size, bytes	Genome Code Letter
64	a
128	b
256	c
512	d
1024	e
1280	f
1518	g
2112	h
9000	i
16000	j
MTU	z

For example: a five packet sequence with sizes 64,64,64,1280,1518 would be designated:

IMIX - aaafg

If z (MTU) is used, the tester MUST specify the length of the MTU in the report.

While this approach allows some flexibility, there are also constraints.

- o Non-RFC2544 packet sizes would need to be approximated by those available in the table.
- o The Genome for very long sequences can become undecipherable by humans.

Some questions testers must ask and answer when using the IMIX Genome are:

1. Multiple Source-Destination Address Pairs: is the IMIX sequence applicable to each pair, across multiple pairs in sets, or across all pairs?
2. Multiple Tester Ports: is the IMIX sequence applicable to each port, across multiple ports in sets, or across all ports?

The chosen configuration would be expressed in the following general form:

Source Address + Port AND/OR Blade	Destination Address + Port AND/OR Blade	Corresponding IMIX
x.x.x.x Blade2	y.y.y.y Blade3	IMIX - aaafg

where testers can specify the IMIX used between any two entities in the test architecture (and Blade is a component in a multi-component device chassis).

4. Specification of a Custom IMIX

This section describes how to specify an IMIX with locally-selected packet sizes

The Custom IMIX is specified in the following format:

CUSTOM IMIX - 123456...x

where each number is replaced by the letter corresponding to the size of the packet at that position in the sequence. The tester MUST complete the following table, giving the letter encoding for each size used, where each set of three lower-case letters would be replaced by the integer size in octets.

Size, bytes	Custom Code Letter
aaa	A
bbb	B
ccc	C
ddd	D
eee	E
fff	F
ggg	G
etc.	up to Z

For example: a five packet sequence with sizes
aaa=64,aaa=64,aaa=64,ggg=1020,ggg=1020 would be designated:

CUSTOM IMIX - AAAGG

5. Reporting Long or Pseudo-Random Packet Sequences

When the IMIX-Genome cannot be used (when the sheer length of the sequence would make the Genome unmanageable), two options are possible. When a sequence can be decomposed into a series of short repeating sequences, then a run-length encoding approach MAY be specified as shown in the table below (using the single lower-case letter Genome Codes from section 3):

Count of Repeating Sequences	Packet Size Sequence
20	abcd
5	ggga
10	dcb

The run-length encoding approach is also applicable to custom IMIX described in section 4 (where the single upper-case letter Genome Codes would be used instead).

When the sequence is designed to vary within some proportional constraints, a table simply giving the proportions of each size MAY be used instead.

IP Length	Percentage of Total	Length(s) at other layers
64	23	82
128	67	146
1000	10	1018

Note that the table of proportions also allows non-standard packet sizes, but trades the short Genome specification and ability to specify the exact sequence for other flexibilities.

If a deterministic packet size generation method is used (such as monotonic increase by one octet from start value to MTU), then the generation algorithm SHOULD be reported.

If a pseudo-random length generation capability is used, then the generation algorithm SHOULD be reported with the results along with the seed value used. We also recognize the opportunity to randomize inter-packet spacing from a test sender as well as the size, and both spacing and length pseudo-random generation algorithms and seeds SHOULD be reported when used.

Finally, we note another possibility: a pseudo-random sequence generates an index to the table of packet lengths, and the generation algorithm SHOULD be reported with the results along with the seed value if used.

6. 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 other constraints [RFC2544].

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.

7. IANA Considerations

This memo makes no requests of IANA, and hopes that IANA will leave it alone as well.

8. Acknowledgements

Thanks to Sarah Banks, Aamer Akhter, Steve Maxwell, and Scott Bradner for their reviews and comments. Ilya Varlashkin suggested the run-length coding approach in Section 5.

9. References

9.1. Normative References

- [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.

9.2. Informative References

- [Agilent] http://www.ixiacom.com/pdfs/test_plans/agilent_journal_of_internet_test_methodologies.pdf, "The Journal of Internet Test Methodologies", 2007.
- [IMIXonList] <http://www.ietf.org/mail-archive/web/bmwg/current/msg00691.html>, "Discussion on IMIX", 2003.
- [IXIA] http://www.ixiacom.com/library/test_plans/display?skey=testing_pppox, "Library: Test Plans", 2010.
- [RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.
- [Spirent] <http://gospirent.com/whitepaper/IMIX%20Test%20Methodolgy%20Journal.pdf>, "Test Methodology Journal: IMIX (Internet Mix) Journal", 2006.
- [jumbo] <http://sd.wareonearth.com/~phil/jumbo.html> and <http://staff.psc.edu/mathis/MTU/arguments.html#crc>, "Discussion of Jumbo Packets and FCS Failure".

Author's Address

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Phone: +1 732 420 1571
Fax: +1 732 368 1192
Email: acmorton@att.com
URI: <http://home.comcast.net/~acmacm/>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 28, 2013

R. Papneja
Huawei Technologies
S. Vapiwala
J. Karthik
Cisco Systems
S. Poretsky
Allot Communications
S. Rao
Qwest Communications
JL. Le Roux
France Telecom
November 29, 2012

Methodology for Benchmarking MPLS-TE Fast Reroute Protection
draft-ietf-bmwg-protection-meth-14.txt

Abstract

This draft describes the methodology for benchmarking MPLS Fast Reroute (FRR) protection mechanisms for link and node protection. This document provides test methodologies and testbed setup for measuring failover times of Fast Reroute techniques while considering factors (such as underlying links) that might impact recovery times for real-time applications bound to MPLS traffic engineered (MPLS-TE) tunnels.

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 May 9, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	5
2. Document Scope	6
3. Existing Definitions and Requirements	6
4. General Reference Topology	7
5. Test Considerations	8
5.1. Failover Events [RFC 6414]	8
5.2. Failure Detection [RFC 6414]	9
5.3. Use of Data Traffic for MPLS Protection benchmarking	10
5.4. LSP and Route Scaling	10
5.5. Selection of IGP	10
5.6. Restoration and Reversion [RFC 6414]	10
5.7. Offered Load	11
5.8. Tester Capabilities	11
5.9. Failover Time Measurement Methods	12
6. Reference Test Setup	12
6.1. Link Protection	13
6.1.1. Link Protection - 1 hop primary (from PLR) and 1 hop backup TE tunnels	13
6.1.2. Link Protection - 1 hop primary (from PLR) and 2 hop backup TE tunnels	14
6.1.3. Link Protection - 2+ hop (from PLR) primary and 1 hop backup TE tunnels	14
6.1.4. Link Protection - 2+ hop (from PLR) primary and 2 hop backup TE tunnels	15
6.2. Node Protection	16
6.2.1. Node Protection - 2 hop primary (from PLR) and 1 hop backup TE tunnels	16
6.2.2. Node Protection - 2 hop primary (from PLR) and 2 hop backup TE tunnels	17
6.2.3. Node Protection - 3+ hop primary (from PLR) and 1 hop backup TE tunnels	18
6.2.4. Node Protection - 3+ hop primary (from PLR) and 2 hop backup TE tunnels	19
7. Test Methodology	20
7.1. MPLS FRR Forwarding Performance	20
7.1.1. Headend PLR Forwarding Performance	20
7.1.2. Mid-Point PLR Forwarding Performance	21
7.2. Headend PLR with Link Failure	23
7.3. Mid-Point PLR with Link Failure	24
7.4. Headend PLR with Node Failure	26
7.5. Mid-Point PLR with Node Failure	27
8. Reporting Format	28
9. Security Considerations	30
10. IANA Considerations	30
11. Acknowledgements	30
12. References	30

12.1. Informative References	30
12.2. Normative References	30
Appendix A. Fast Reroute Scalability Table	30
Appendix B. Abbreviations	33
Authors' Addresses	34

1. Introduction

This document describes the methodology for benchmarking MPLS Fast Reroute (FRR) protection mechanisms. This document uses much of the terminology defined in [RFC 6414].

Protection mechanisms provide recovery of client services from a planned or an unplanned link or node failures. MPLS FRR protection mechanisms are generally deployed in a network infrastructure where MPLS is used for provisioning of point-to-point traffic engineered tunnels (tunnel). MPLS FRR protection mechanisms aim to reduce service disruption period by minimizing recovery time from most common failures.

Network elements from different manufacturers behave differently to network failures, which impacts the network's ability and performance for failure recovery. It therefore becomes imperative for service providers to have a common benchmark to understand the performance behaviors of network elements.

There are two factors impacting service availability: frequency of failures and duration for which the failures persist. Failures can be classified further into two types: correlated and uncorrelated. Correlated and uncorrelated failures may be planned or unplanned.

Planned failures are generally predictable. Network implementations should be able to handle both planned and unplanned failures and recover gracefully within a time frame to maintain service assurance. Hence, failover recovery time is one of the most important benchmark that a service provider considers in choosing the building blocks for their network infrastructure.

A correlated failure is a result of the occurrence of two or more failures. A typical example is failure of a logical resource (e.g. layer-2 links) due to a dependency on a common physical resource (e.g. common conduit) that fails. Within the context of MPLS protection mechanisms, failures that arise due to Shared Risk Link Groups (SRLG) [RFC 4202] can be considered as correlated failures.

MPLS FRR [RFC 4090] allows for the possibility that the Label Switched Paths can be re-optimized in the minutes following Failover. IP Traffic would be re-routed according to the preferred path for the post-failure topology. Thus, MPLS-FRR may include additional steps following the occurrence of the failure detection [RFC 6414] and failover event [RFC 6414].

- (1) Failover Event - Primary Path (Working Path) fails
- (2) Failure Detection- Failover Event is detected
- (3)
 - a. Failover - Working Path switched to Backup path
 - b. Re-Optimization of Working Path (possible change from Backup Path)
- (4) Restoration [RFC 6414]
- (5) Reversion [RFC 6414]

2. Document Scope

This document provides detailed test cases along with different topologies and scenarios that should be considered to effectively benchmark MPLS FRR protection mechanisms and failover times on the Data Plane. Different Failover Events and scaling considerations are also provided in this document.

All benchmarking test-cases defined in this document apply to Facility backup [RFC 4090]. The test cases cover set of interesting failure scenarios and the associated procedures benchmark the performance of the Device Under Test (DUT) to recover from failures. Data plane traffic is used to benchmark failover times. Testing scenarios related to MPLS-TE protection mechanisms when applied to MPLS Transport Profile and IP fast reroute applied to MPLS networks were not considered and are out of scope of this document. However, the test setups considered for MPLS based Layer 3 and Layer 2 services consider LDP over MPLS RSVP-TE configurations.

Benchmarking of correlated failures is out of scope of this document. Detection using Bi-directional Forwarding Detection (BFD) is outside the scope of this document, but mentioned in discussion sections.

The Performance of control plane is outside the scope of this benchmarking.

As described above, MPLS-FRR may include a Re-optimization of the Working Path, with possible packet transfer impairments. Characterization of Re-optimization is beyond the scope of this memo.

3. Existing Definitions and Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

The tester MUST record the number of lost, duplicate, and out-of-order packets. It should further record arrival and departure times so that Failover Time, Additive Latency, and Reversion Time can be measured. The tester may be a single device or a test system emulating all the different roles along a primary or backup path.

The label stack is dependent of the following 3 entities:

- (1) Type of protection (Link Vs Node)
- (2) # of remaining hops of the primary tunnel from the PLR[RFC 6414]
- (3) # of remaining hops of the backup tunnel from the PLR

Due to this dependency, it is RECOMMENDED that the benchmarking of failover times be performed on all the topologies provided in section 6.

5. Test Considerations

This section discusses the fundamentals of MPLS Protection testing:

- (1) The types of network events that causes failover (section 5.1)
- (2) Indications for failover (section 5.2)
- (3) the use of data traffic (section 5.3)
- (4) LSP Scaling (Section 5.4)
- (5) IGP Selection (Section 5.5)
- (6) Reversion of LSP (Section 5.6)
- (7) Traffic generation (section 5.7)

5.1. Failover Events [RFC 6414]

The failover to the backup tunnel is primarily triggered by either link or node failures observed downstream of the Point of Local repair (PLR). The failure events are listed below.

Link Failure Events

- Interface Shutdown on PLR side with physical/link Alarm
- Interface Shutdown on remote side with physical/link Alarm
- Interface Shutdown on PLR side with RSVP hello enabled
- Interface Shutdown on remote side with RSVP hello enabled
- Interface Shutdown on PLR side with BFD
- Interface Shutdown on remote side with BFD
- Fiber Pull on the PLR side (Both TX & RX or just the TX)
- Fiber Pull on the remote side (Both TX & RX or just the RX)
- Online insertion and removal (OIR) on PLR side
- OIR on remote side
- Sub-interface failure on PLR side (e.g. shutting down of a VLAN)
- Sub-interface failure on remote side
- Parent interface shutdown on PLR side (an interface bearing multiple sub-interfaces)
- Parent interface shutdown on remote side

Node Failure Events

- A System reload initiated either by a graceful shutdown or by a power failure.
- A system crash due to a software failure or an assert.

5.2. Failure Detection [RFC 6414]

Link failure detection time depends on the link type and failure detection protocols running. For SONET/SDH, the alarm type (such as LOS, AIS, or RDI) can be used. Other link types have layer-two alarms, but they may not provide a short enough failure detection time. Ethernet based links enabled with MPLS/IP do not have layer 2 failure indicators, and therefore relies on layer 3 signaling for failure detection. However for directly connected devices, remote fault indication in the ethernet auto-negotiation scheme could be considered as a type of layer 2 link failure indicator.

MPLS has different failure detection techniques such as BFD, or use of RSVP hellos. These methods can be used for the layer 3 failure indicators required by Ethernet based links, or for some other non-Ethernet based links to help improve failure detection time. However, these fast failure detection mechanisms are out of scope.

The test procedures in this document can be used for a local failure or remote failure scenarios for comprehensive benchmarking and to evaluate failover performance independent of the failure detection techniques.

5.3. Use of Data Traffic for MPLS Protection benchmarking

Currently end customers use packet loss as a key metric for Failover Time [RFC 6414]. Failover Packet Loss [RFC 6414] is an externally observable event and has direct impact on application performance. MPLS protection is expected to minimize the packet loss in the event of a failure. For this reason it is important to develop a standard router benchmarking methodology for measuring MPLS protection that uses packet loss as a metric. At a known rate of forwarding, packet loss can be measured and the failover time can be determined. Measurement of control plane signaling to establish backup paths is not enough to verify failover. Failover is best determined when packets are actually traversing the backup path.

An additional benefit of using packet loss for calculation of failover time is that it allows use of a black-box test environment. Data traffic is offered at line-rate to the device under test (DUT) an emulated network failure event is forced to occur, and packet loss is externally measured to calculate the convergence time. This setup is independent of the DUT architecture.

In addition, this methodology considers the packets in error and duplicate packets [RFC 4689] that could have been generated during the failover process. The methodologies consider lost, out-of-order [RFC 4689] and duplicate packets to be impaired packets that contribute to the Failover Time.

5.4. LSP and Route Scaling

Failover time performance may vary with the number of established primary and backup tunnel label switched paths (LSP) and installed routes. However the procedure outlined here should be used for any number of LSPs (L) and number of routes protected by PLR(R). The amount of L and R must be recorded.

5.5. Selection of IGP

The underlying IGP could be ISIS-TE or OSPF-TE for the methodology proposed here. See [RFC 6412] for IGP options to consider and report.

5.6. Restoration and Reversion [RFC 6414]

Path restoration provides a method to restore an alternate primary LSP upon failure and to switch traffic from the Backup Path to the restored Primary Path (Reversion). In MPLS-FRR, Reversion can be implemented as Global Reversion or Local Reversion. It is important to include Restoration and Reversion as a step in each test case to

measure the amount of packet loss, out of order packets, or duplicate packets that is produced.

Note: In addition to restoration and reversion, re-optimization can take place while the failure is still not recovered but it depends on the user configuration, and re-optimization timers.

5.7. Offered Load

It is suggested that there be three or more traffic streams as long as there is a steady and constant rate of flow for all the streams. In order to monitor the DUT performance for recovery times, a set of route prefixes should be advertised before traffic is sent. The traffic should be configured towards these routes.

Prefix-dependency behaviors are key in IP and tests with route-specific flows spread across the routing table will reveal this dependency. Generating traffic to all of the prefixes reachable by the protected tunnel (probably in a Round-Robin fashion, where the traffic is destined to all the prefixes but one prefix at a time in a cyclic manner) is not recommended. Round-Robin traffic generation is not recommended to all prefixes, as time to hit all the prefixes may be higher than the failover time. This phenomenon will reduce the granularity of the measured results and the results observed may not be accurate.

5.8. Tester Capabilities

It is RECOMMENDED that the Tester used to execute each test case have the following capabilities:

- 1.Ability to establish MPLS-TE tunnels and push/pop labels.
- 2.Ability to produce Failover Event [RFC 6414].
- 3.Ability to insert a timestamp in each data packet's IP payload.
- 4.An internal time clock to control timestamping, time measurements, and time calculations.
- 5.Ability to disable or tune specific Layer-2 and Layer-3 protocol functions on any interface(s).

6. Ability to react upon the receipt of path error from the PLR

The Tester MAY be capable to make non-data plane convergence observations and use those observations for measurements.

5.9. Failover Time Measurement Methods

Failover Time is calculated using one of the following three methods

1. Packet-Loss Based method (PLBM): (Number of packets dropped/ packets per second * 1000) milliseconds. This method could also be referred as Loss-Derived method.
2. Time-Based Loss Method (TBLM): This method relies on the ability of the Traffic generators to provide statistics which reveal the duration of failure in milliseconds based on when the packet loss occurred (interval between non-zero packet loss and zero loss).
3. Timestamp Based Method (TBM): This method of failover calculation is based on the timestamp that gets transmitted as payload in the packets originated by the generator. The Traffic Analyzer records the timestamp of the last packet received before the failover event and the first packet after the failover and derives the time based on the difference between these 2 timestamps. Note: The payload could also contain sequence numbers for out-of-order packet calculation and duplicate packets.

The timestamp based method would be able to detect Reversion impairments beyond loss, thus it is RECOMMENDED method as a Failover Time method.

6. Reference Test Setup

In addition to the general reference topology shown in figure 1, this section provides detailed insight into various proposed test setups that should be considered for comprehensively benchmarking the failover time in different roles along the primary tunnel

This section proposes a set of topologies that covers all the scenarios for local protection. All of these topologies can be mapped to the reference topology shown in Figure 1. Topologies provided in this section refer to the testbed required to benchmark failover time when the DUT is configured as a PLR in either Headend or midpoint role. Provided with each topology below is the label stack at the PLR. Penultimate Hop Popping (PHP) MAY be used and must be reported when used.

Figures 2 thru 9 use the following convention and are subset of figure 1:

- a) HE is Headend
- b) TE is Tail-End
- c) MID is Mid point
- d) MP is Merge Point
- e) PLR is Point of Local Repair
- f) PRI is Primary Path
- g) BKP denotes Backup Path and Nodes
- h) UR is Upstream Router

6.1. Link Protection

6.1.1. Link Protection - 1 hop primary (from PLR) and 1 hop backup TE tunnels

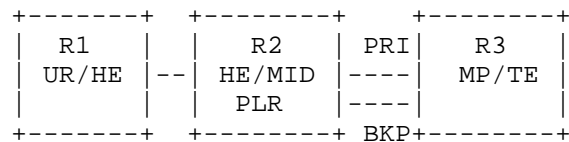


Figure 2.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	0	0
Layer3 VPN (PE-PE)	1	1
Layer3 VPN (PE-P)	2	2
Layer2 VC (PE-PE)	1	1
Layer2 VC (PE-P)	2	2
Mid-point LSPs	0	0

Note: Please note the following:

- a) For P-P case, R2 and R3 acts as P routers
- b) For PE-PE case, R2 acts as PE and R3 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R3 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2 and R3 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.1.2. Link Protection - 1 hop primary (from PLR) and 2 hop backup TE tunnels

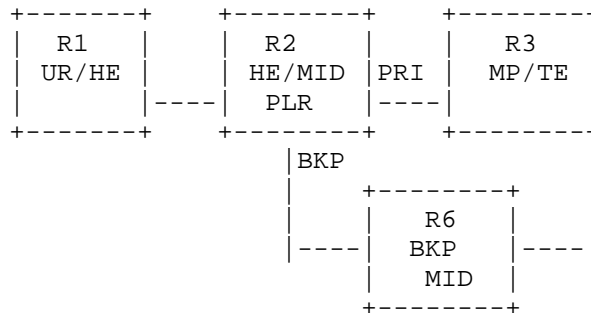


Figure 3.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	0	1
Layer3 VPN (PE-PE)	1	2
Layer3 VPN (PE-P)	2	3
Layer2 VC (PE-PE)	1	2
Layer2 VC (PE-P)	2	3
Mid-point LSPs	0	1

Note: Please note the following:

- a) For P-P case, R2 and R3 acts as P routers
- b) For PE-PE case, R2 acts as PE and R3 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R3 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2 and R3 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.1.3. Link Protection - 2+ hop (from PLR) primary and 1 hop backup TE tunnels

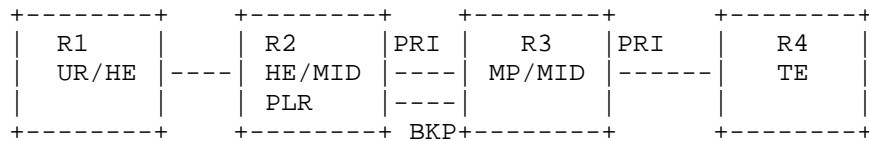


Figure 4.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	1
Layer3 VPN (PE-PE)	2	2
Layer3 VPN (PE-P)	3	3
Layer2 VC (PE-PE)	2	2
Layer2 VC (PE-P)	3	3
Mid-point LSPs	1	1

Note: Please note the following:

- a) For P-P case, R2, R3 and R4 acts as P routers
- b) For PE-PE case, R2 acts as PE and R4 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R3 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3 and R4 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.1.4. Link Protection - 2+ hop (from PLR) primary and 2 hop backup TE tunnels

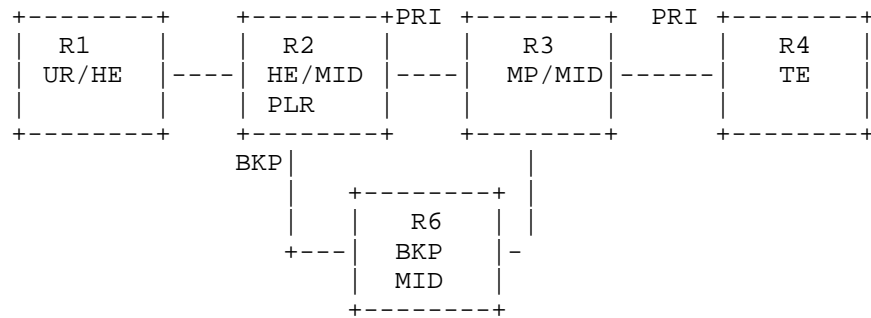


Figure 5.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	2
Layer3 VPN (PE-PE)	2	3
Layer3 VPN (PE-P)	3	4
Layer2 VC (PE-PE)	2	3
Layer2 VC (PE-P)	3	4
Mid-point LSPs	1	2

Note: Please note the following:

- a) For P-P case, R2, R3 and R4 acts as P routers
- b) For PE-PE case, R2 acts as PE and R4 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R3 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3 and R4 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.2. Node Protection

6.2.1. Node Protection - 2 hop primary (from PLR) and 1 hop backup TE tunnels

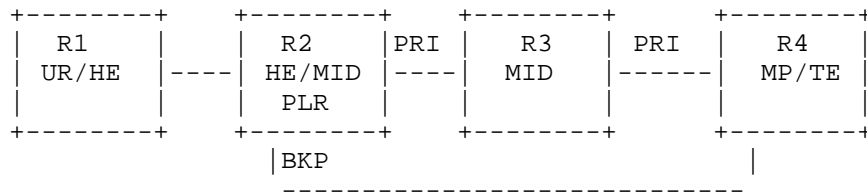


Figure 6.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	0
Layer3 VPN (PE-PE)	2	1
Layer3 VPN (PE-P)	3	2
Layer2 VC (PE-PE)	2	1
Layer2 VC (PE-P)	3	2
Mid-point LSPs	1	0

Note: Please note the following:

- a) For P-P case, R2, R3 and R3 acts as P routers
- b) For PE-PE case, R2 acts as PE and R4 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R4 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3 and R4 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.2.2. Node Protection - 2 hop primary (from PLR) and 2 hop backup TE tunnels

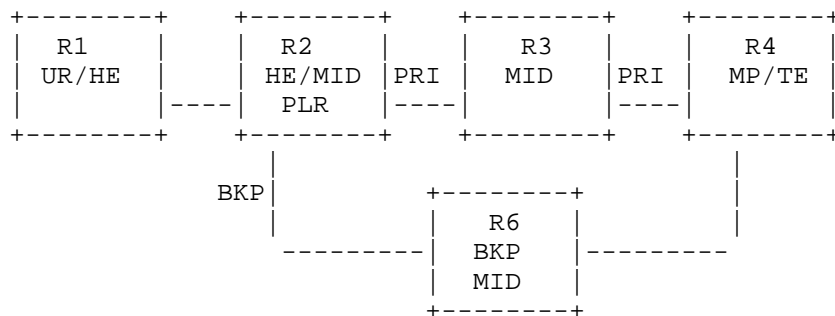


Figure 7.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	1
Layer3 VPN (PE-PE)	2	2
Layer3 VPN (PE-P)	3	3
Layer2 VC (PE-PE)	2	2
Layer2 VC (PE-P)	3	3
Mid-point LSPs	1	1

Note: Please note the following:

- a) For P-P case, R2, R3 and R4 acts as P routers
- b) For PE-PE case, R2 acts as PE and R4 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R4 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3 and R4 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.2.3. Node Protection - 3+ hop primary (from PLR) and 1 hop backup TE tunnels

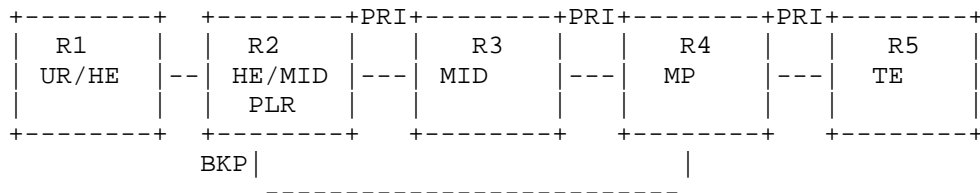


Figure 8.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	1
Layer3 VPN (PE-PE)	2	2
Layer3 VPN (PE-P)	3	3
Layer2 VC (PE-PE)	2	2
Layer2 VC (PE-P)	3	3
Mid-point LSPs	1	1

Note: Please note the following:

- a) For P-P case, R2, R3, R4 and R5 acts as P routers
- b) For PE-PE case, R2 acts as PE and R5 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R4 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3, R4 and R5 act as shown in above figure HE, Midpoint/PLR and TE respectively

6.2.4. Node Protection - 3+ hop primary (from PLR) and 2 hop backup TE tunnels

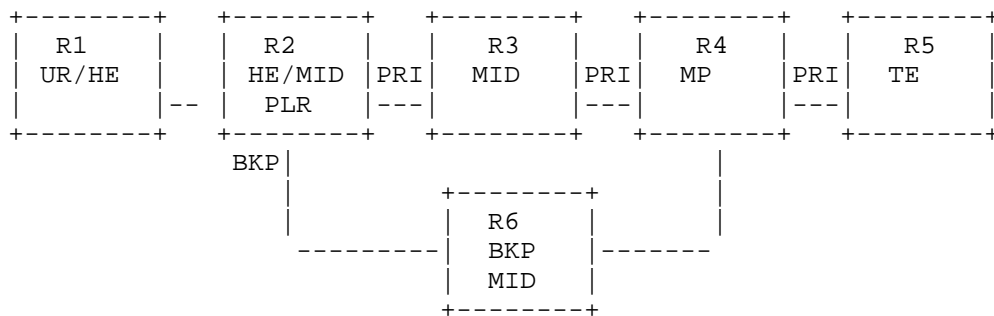


Figure 9.

Traffic	Num of Labels before failure	Num of labels after failure
IP TRAFFIC (P-P)	1	2
Layer3 VPN (PE-PE)	2	3
Layer3 VPN (PE-P)	3	4
Layer2 VC (PE-PE)	2	3
Layer2 VC (PE-P)	3	4
Mid-point LSPs	1	2

Note: Please note the following:

- a) For P-P case, R2, R3, R4 and R5 acts as P routers
- b) For PE-PE case, R2 acts as PE and R5 acts as a remote PE
- c) For PE-P case, R2 acts as a PE router, R4 acts as a P router and R5 acts as remote PE router (Please refer to figure 1 for complete setup)
- d) For Mid-point case, R1, R2, R3, R4 and R5 act as shown in above figure HE, Midpoint/PLR and TE respectively

7. Test Methodology

The procedure described in this section can be applied to all the 8 base test cases and the associated topologies. The backup as well as the primary tunnels are configured to be alike in terms of bandwidth usage. In order to benchmark failover with all possible label stack depth applicable as seen with current deployments, it is RECOMMENDED to perform all of the test cases provided in this section. The forwarding performance test cases in section 7.1 MUST be performed prior to performing the failover test cases.

The considerations of Section 4 of [RFC 2544] are applicable when evaluating the results obtained using these methodologies as well.

7.1. MPLS FRR Forwarding Performance

Benchmarking Failover Time [RFC 6414] for MPLS protection first requires baseline measurement of the forwarding performance of the test topology including the DUT. Forwarding performance is benchmarked by the Throughput as defined in [RFC 5695] and measured in units pps. This section provides two test cases to benchmark forwarding performance. These are with the DUT configured as a Headend PLR, Mid-Point PLR, and Egress PLR.

7.1.1. Headend PLR Forwarding Performance

Objective:

To benchmark the maximum rate (pps) on the PLR (as headend) over primary LSP and backup LSP.

Test Setup:

- A. Select any one topology out of the 8 from section 6.
- B. Select or enable IP, Layer 3 VPN or Layer 2 VPN services with DUT as Headend PLR.
- C. The DUT will also have 2 interfaces connected to the traffic Generator/analyzer. (If the node downstream of the PLR is not a simulated node, then the Ingress of the tunnel should have one link connected to the traffic generator and the node downstream to the PLR or the egress of the tunnel should have a link connected to the traffic analyzer).

Procedure:

1. Establish the primary LSP on R2 required by the topology selected.
2. Establish the backup LSP on R2 required by the selected topology.
3. Verify primary and backup LSPs are up and that primary is protected.
4. Verify Fast Reroute protection is enabled and ready.
5. Setup traffic streams as described in section 5.7.
6. Send MPLS traffic over the primary LSP at the Throughput supported by the DUT (section 6, RFC 2544).
7. Record the Throughput over the primary LSP.
8. Trigger a link failure as described in section 5.1.
9. Verify that the offered load gets mapped to the backup tunnel and measure the Additive Backup Delay (RFC 6414).
10. 30 seconds after Failover, stop the offered load and measure the Throughput, Packet Loss, Out-of-Order Packets, and Duplicate Packets over the Backup LSP.
11. Adjust the offered load and repeat steps 6 through 10 until the Throughput values for the primary and backup LSPs are equal.
12. Record the final Throughput, which corresponds to the offered load that will be used for the Headend PLR failover test cases.

7.1.2. Mid-Point PLR Forwarding Performance

Objective:

To benchmark the maximum rate (pps) on the PLR (as mid-point) over primary LSP and backup LSP.

Test Setup:

- A. Select any one topology out of the 8 from section 6.
- B. The DUT will also have 2 interfaces connected to the traffic generator.

Procedure:

1. Establish the primary LSP on R1 required by the topology selected.
2. Establish the backup LSP on R2 required by the selected topology.
3. Verify primary and backup LSPs are up and that primary is protected.
4. Verify Fast Reroute protection is enabled and ready.
5. Setup traffic streams as described in section 5.7.
6. Send MPLS traffic over the primary LSP at the Throughput supported by the DUT (section 6, RFC 2544).
7. Record the Throughput over the primary LSP.
8. Trigger a link failure as described in section 5.1.
9. Verify that the offered load gets mapped to the backup tunnel and measure the Additive Backup Delay (RFC 6414).
10. 30 seconds after Failover, stop the offered load and measure the Throughput, Packet Loss, Out-of-Order Packets, and Duplicate Packets over the Backup LSP.
11. Adjust the offered load and repeat steps 6 through 10 until the Throughput values for the primary and backup LSPs are equal.
12. Record the final Throughput which corresponds to the offered load that will be used for the Mid-Point PLR failover test cases.

7.2. Headend PLR with Link Failure

Objective:

To benchmark the MPLS failover time due to link failure events described in section 5.1 experienced by the DUT which is the Headend PLR.

Test Setup:

- A. Select any one topology out of the 8 from section 6.
- B. Select or enable IP, Layer 3 VPN or Layer 2 VPN services with DUT as Headend PLR.
- C. The DUT will also have 2 interfaces connected to the traffic Generator/analyzer. (If the node downstream of the PLR is not a simulated node, then the Ingress of the tunnel should have one link connected to the traffic generator and the node downstream to the PLR or the egress of the tunnel should have a link connected to the traffic analyzer).

Test Configuration:

1. Configure the number of primaries on R2 and the backups on R2 as required by the topology selected.
2. Configure the test setup to support Reversion.
3. Advertise prefixes (as per FRR Scalability Table described in Appendix A) by the tail end.

Procedure:

Test Case "7.1.1. Headend PLR Forwarding Performance" MUST be completed first to obtain the Throughput to use as the offered load.

1. Establish the primary LSP on R2 required by the topology selected.

2. Establish the backup LSP on R2 required by the selected topology.
3. Verify primary and backup LSPs are up and that primary is protected.
4. Verify Fast Reroute protection is enabled and ready.
5. Setup traffic streams for the offered load as described in section 5.7.
6. Provide the offered load from the tester at the Throughput [RFC 1242] level obtained from test case 7.1.1.
7. Verify traffic is switched over Primary LSP without packet loss.
8. Trigger a link failure as described in section 5.1.
9. Verify that the offered load gets mapped to the backup tunnel and measure the Additive Backup Delay.
10. 30 seconds after Failover [RFC 6414], stop the offered load and measure the total Failover Packet Loss [RFC 6414].
11. Calculate the Failover Time [RFC 6414] benchmark using the selected Failover Time Calculation Method (TBLM, PLBM, or TBM) [RFC 6414].
12. Restart the offered load and restore the primary LSP to verify Reversion [RFC 6414] occurs and measure the Reversion Packet Loss [RFC 6414].
13. Calculate the Reversion Time [RFC 6414] benchmark using the selected Failover Time Calculation Method (TBLM, PLBM, or TBM) [RFC 6414].
14. Verify Headend signals new LSP and protection should be in place again.

IT is RECOMMENDED that this procedure be repeated for each of the link failure triggers defined in section 5.1.

7.3. Mid-Point PLR with Link Failure

Objective:

To benchmark the MPLS failover time due to link failure events described in section 5.1 experienced by the DUT which is the Mid-Point PLR.

Test Setup:

- A. Select any one topology out of the 8 from section 6.
- B. The DUT will also have 2 interfaces connected to the traffic generator.

Test Configuration:

1. Configure the number of primaries on R1 and the backups on R2 as required by the topology selected.
2. Configure the test setup to support Reversion.
3. Advertise prefixes (as per FRR Scalability Table described in Appendix A) by the tail end.

Procedure:

Test Case "7.1.2. Mid-Point PLR Forwarding Performance" MUST be completed first to obtain the Throughput to use as the offered load.

1. Establish the primary LSP on R1 required by the topology selected.
2. Establish the backup LSP on R2 required by the selected topology.
3. Perform steps 3 through 14 from section 7.2 Headend PLR with Link Failure.

IT is RECOMMENDED that this procedure be repeated for each of the link failure triggers defined in section 5.1.

7.4. Headend PLR with Node Failure

Objective:

To benchmark the MPLS failover time due to Node failure events described in section 5.1 experienced by the DUT which is the Headend PLR.

Test Setup:

- A. Select any one topology out of the 8 from section 6.
- B. Select or enable IP, Layer 3 VPN or Layer 2 VPN services with DUT as Headend PLR.
- C. The DUT will also have 2 interfaces connected to the traffic generator/analyzer.

Test Configuration:

1. Configure the number of primaries on R2 and the backups on R2 as required by the topology selected.
2. Configure the test setup to support Reversion.
3. Advertise prefixes (as per FRR Scalability Table described in Appendix A) by the tail end.

Procedure:

Test Case "7.1.1. Headend PLR Forwarding Performance" MUST be completed first to obtain the Throughput to use as the offered load.

1. Establish the primary LSP on R2 required by the topology selected.
2. Establish the backup LSP on R2 required by the selected topology.
3. Verify primary and backup LSPs are up and that primary is protected.

4. Verify Fast Reroute protection is enabled and ready.
5. Setup traffic streams for the offered load as described in section 5.7.
6. Provide the offered load from the tester at the Throughput [RFC 1242] level obtained from test case 7.1.1.
7. Verify traffic is switched over Primary LSP without packet loss.
8. Trigger a node failure as described in section 5.1.
9. Perform steps 9 through 14 in 7.2 Headend PLR with Link Failure.

IT is RECOMMENDED that this procedure be repeated for each of the node failure triggers defined in section 5.1.

7.5. Mid-Point PLR with Node Failure

Objective:

To benchmark the MPLS failover time due to Node failure events described in section 5.1 experienced by the DUT which is the Mid-Point PLR.

Test Setup:

- A. Select any one topology from section 6.1 to 6.2.
- B. The DUT will also have 2 interfaces connected to the traffic generator.

Test Configuration:

1. Configure the number of primaries on R1 and the backups on R2 as required by the topology selected.
2. Configure the test setup to support Reversion.
3. Advertise prefixes (as per FRR Scalability Table described in Appendix A) by the tail end.

Procedure:

Test Case "7.1.1. Mid-Point PLR Forwarding Performance" MUST be completed first to obtain the Throughput to use as the offered load.

1. Establish the primary LSP on R1 required by the topology selected.
2. Establish the backup LSP on R2 required by the selected topology.
3. Verify primary and backup LSPs are up and that primary is protected.
4. Verify Fast Reroute protection is enabled and ready.
5. Setup traffic streams for the offered load as described in section 5.7.
6. Provide the offered load from the tester at the Throughput [RFC 1242] level obtained from test case 7.1.1.
7. Verify traffic is switched over Primary LSP without packet loss.
8. Trigger a node failure as described in section 5.1.
9. Perform steps 9 through 14 in 7.2 Headend PLR with Link Failure.

IT is RECOMMENDED that this procedure be repeated for each of the node failure triggers defined in section 5.1.

8. Reporting Format

For each test, it is RECOMMENDED that the results be reported in the following format.

Parameter	Units
IGP used for the test	ISIS-TE/ OSPF-TE

Interface types	Gige,POS,ATM,VLAN etc.
Packet Sizes offered to the DUT	Bytes (at layer 3)
Offered Load (Throughput)	packets per second
IGP routes advertised	Number of IGP routes
Penultimate Hop Popping	Used/Not Used
RSVP hello timers	Milliseconds
Number of Protected tunnels	Number of tunnels
Number of VPN routes installed on the Headend	Number of VPN routes
Number of VC tunnels	Number of VC tunnels
Number of mid-point tunnels	Number of tunnels
Number of Prefixes protected by Primary	Number of LSPs
Topology being used	Section number, and figure reference
Failover Event	Event type
Re-optimization	Yes/No
Benchmarks (to be recorded for each test case):	
Failover-	
Failover Time	seconds
Failover Packet Loss	packets
Additive Backup Delay	seconds
Out-of-Order Packets	packets
Duplicate Packets	packets
Failover Time Calculation Method	Method Used
Reversion-	
Reversion Time	seconds
Reversion Packet Loss	packets
Additive Backup Delay	seconds
Out-of-Order Packets	packets
Duplicate Packets	packets
Failover Time Calculation Method	Method Used

9. 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.

10. IANA Considerations

This draft does not require any new allocations by IANA.

11. Acknowledgements

We would like to thank Jean Philip Vasseur for his invaluable input to the document, Curtis Villamizar for his contribution in suggesting text on definition and need for benchmarking Correlated failures and Bhavani Parise for his textual input and review. Additionally we would like to thank Al Morton, Arun Gandhi, Amrit Hanspal, Karu Ratnam, Raveesh Janardan, Andrey Kiselev, and Mohan Nanduri for their formal reviews of this document.

12. References

12.1. Informative References

- [RFC 2285] Mandeville, R., "Benchmarking Terminology for LAN Switching Devices", RFC 2285, February 1998.
- [RFC 4689] Poretsky, S., Perser, J., Erramilli, S., and S. Khurana, "Terminology for Benchmarking Network-layer Traffic Control Mechanisms", RFC 4689, October 2006.
- [RFC 4202] Kompella, K., Rekhter, Y., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

12.2. Normative References

- [RFC 1242] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [RFC 4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC 5695] Akhter, A., Asati, R., and C. Pignataro, "MPLS Forwarding Benchmarking Methodology for IP Flows", RFC 5695, November 2009.
- [RFC 6414] Poretsky, S., Papneja, R., Karthik, J., and S. Vapiwala, "Benchmarking Terminology for Protection Performance", RFC 6414, November 2011.
- [RFC 2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.
- [RFC 6412] Poretsky, S., Imhoff, B., and K. Michielsen, "Terminology for Benchmarking Link-State IGP Data-Plane Route Convergence", RFC 6412, November 2011.

Appendix A. Fast Reroute Scalability Table

This section provides the recommended numbers for evaluating the scalability of fast reroute implementations. It also recommends the typical numbers for IGP/VPNv4 Prefixes, LSP Tunnels and VC entries. Based on the features supported by the device under test (DUT), appropriate scaling limits can be used for the test bed.

A1. FRR IGP Table

No. of Headend TE Tunnels	IGP Prefixes
1	100
1	500
1	1000
1	2000
1	5000
2 (Load Balance)	100
2 (Load Balance)	500
2 (Load Balance)	1000
2 (Load Balance)	2000
2 (Load Balance)	5000
100	100
500	500
1000	1000
2000	2000

A2. FRR VPN Table

No. of Headend TE Tunnels	VPNv4 Prefixes
1	100
1	500
1	1000
1	2000
1	5000
1	10000
1	20000
1	Max
2 (Load Balance)	100
2 (Load Balance)	500
2 (Load Balance)	1000
2 (Load Balance)	2000
2 (Load Balance)	5000
2 (Load Balance)	10000
2 (Load Balance)	20000
2 (Load Balance)	Max

A3. FRR Mid-Point LSP Table

No of Mid-point TE LSPs could be configured at recommended levels - 100, 500, 1000, 2000, or max supported number.

A2. FRR VC Table

No. of Headend TE Tunnels	VC entries
1	100
1	500
1	1000
1	2000
1	Max
100	100
500	500
1000	1000
2000	2000

Appendix B. Abbreviations

AIS	- Alarm Indication Signal
BFD	- Bidirectional Fault Detection
BGP	- Border Gateway protocol
CE	- Customer Edge
DUT	- Device Under Test
FRR	- Fast Reroute
IGP	- Interior Gateway Protocol
IP	- Internet Protocol
LOS	- Loss of Signal
LSP	- Label Switched Path
MP	- Merge Point
MPLS	- Multi Protocol Label Switching
N-Nhop	- Next - Next Hop
Nhop	- Next Hop
OIR	- Online Insertion and Removal
P	- Provider
PE	- Provider Edge
PHP	- Penultimate Hop Popping
PLR	- Point of Local Repair
RSVP	- Resource reSerVation Protocol
SRLG	- Shared Risk Link Group
TA	- Traffic Analyzer
TE	- Traffic Engineering
TG	- Traffic Generator
VC	- Virtual Circuit
VPN	- Virtual Private Network

Authors' Addresses

Rajiv Papneja
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: rajiv.papneja@huawei.com

Samir Vapiwala
Cisco Systems
300 Beaver Brook Road
Boxborough, MA 01719
USA

Email: svapiwal@cisco.com

Jay Karthik
Cisco Systems
300 Beaver Brook Road
Boxborough, MA 01719
USA

Email: jkarthik@cisco.com

Scott Poretsky
Allot Communications
USA

Email: sporetsky@allot.com

Shankar Rao
Qwest Communications
950 17th Street
Suite 1900
Denver, CO 80210
USA

Email: shankar.rao@du.edu

JL. Le Roux
France Telecom
2 av Pierre Marzin
22300 Lannion
France

Email: jeanlouis.leroux@orange.com

Benchmarking Methodology Working
Group
Internet-Draft
Intended status: Informational
Expires: July 12, 2013

C. Davids
Illinois Institute of Technology
V. Gurbani
Bell Laboratories, Alcatel-Lucent
S. Poretsky
Allot Communications
January 8, 2013

Methodology for Benchmarking SIP Networking Devices
draft-ietf-bmwg-sip-bench-meth-08

Abstract

This document describes the methodology for benchmarking Session Initiation Protocol (SIP) performance as described in SIP benchmarking terminology document. The methodology and terminology are to be used for benchmarking signaling plane performance with varying signaling and media load. Both scale and establishment rate are measured by signaling plane performance. The SIP Devices to be benchmarked may be a single device under test (DUT) or a system under test (SUT). Benchmarks can be obtained and compared for different types of devices such as SIP Proxy Server, SBC, and server paired with a media relay or Firewall/NAT device.

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 July 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents
(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminology	4
2. Introduction	4
3. Benchmarking Topologies	5
4. Test Setup Parameters	5
4.1. Selection of SIP Transport Protocol	5
4.2. Signaling Server	5
4.3. Associated Media	5
4.4. Selection of Associated Media Protocol	6
4.5. Number of Associated Media Streams per SIP Session	6
4.6. Session Duration	6
4.7. Attempted Sessions per Second	6
4.8. Stress Testing	6
4.9. Benchmarking algorithm	6
5. Reporting Format	9
5.1. Test Setup Report	9
5.2. Device Benchmarks for IS	10
5.3. Device Benchmarks for NS	10
6. Test Cases	10
6.1. Baseline Session Establishment Rate of the test bed	10
6.2. Session Establishment Rate without media	11
6.3. Session Establishment Rate with Media not on DUT/SUT	11
6.4. Session Establishment Rate with Media on DUT/SUT	12
6.5. Session Establishment Rate with Loop Detection Enabled	13
6.6. Session Establishment Rate with Forking	13
6.7. Session Establishment Rate with Forking and Loop Detection	14
6.8. Session Establishment Rate with TLS Encrypted SIP	14
6.9. Session Establishment Rate with IPsec Encrypted SIP	15
6.10. Session Establishment Rate with SIP Flooding	16
6.11. Maximum Registration Rate	16
6.12. Maximum Re-Registration Rate	16
6.13. Maximum IM Rate	17
6.14. Session Capacity without Media	17
6.15. Session Capacity with Media	18
6.16. Session Capacity with Media and a Media Relay/NAT and/or Firewall	19
7. IANA Considerations	19
8. Security Considerations	19
9. Acknowledgments	19
10. References	20
10.1. Normative References	20
10.2. Informative References	20
Authors' Addresses	20

1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, conforming to [RFC2119] and indicate requirement levels for compliant implementations.

Terms specific to SIP [RFC3261] performance benchmarking are defined in [I-D.sip-bench-term].

RFC 2119 defines the use of these key words to help make the intent of standards track documents as clear as possible. While this document uses these keywords, this document is not a standards track document. The term Throughput is defined in [RFC2544].

2. Introduction

This document describes the methodology for benchmarking Session Initiation Protocol (SIP) performance as described in Terminology document [I-D.sip-bench-term]. The methodology and terminology are to be used for benchmarking signaling plane performance with varying signaling and media load. Both scale and establishment rate are measured by signaling plane performance.

The SIP Devices to be benchmarked may be a single device under test (DUT) or a system under test (SUT). The DUT is a SIP Server, which may be any [RFC3261] conforming device. The SUT can be any device or group of devices containing RFC 3261 conforming functionality along with Firewall and/or NAT functionality. This enables benchmarks to be obtained and compared for different types of devices such as SIP Proxy Server, SBC, SIP proxy server paired with a media relay or Firewall/NAT device. SIP Associated Media benchmarks can also be made when testing SUTs.

The test cases provide benchmarks metrics of Registration Rate, SIP Session Establishment Rate, Session Capacity, and IM Rate. These can be benchmarked with or without associated Media. Some cases are also included to cover Forking, Loop detection, Encrypted SIP, and SIP Flooding. The test topologies that can be used are described in the Test Setup section. Topologies are provided for benchmarking of a DUT or SUT. Benchmarking with Associated Media can be performed when using a SUT.

SIP permits a wide range of configuration options that are explained in Section 4 and Section 2 of [I-D.sip-bench-term]. Benchmark metrics could possibly be impacted by Associated Media. The selected

values for Session Duration and Media Streams per Session enable benchmark metrics to be benchmarked without Associated Media. Session Setup Rate could possibly be impacted by the selected value for Maximum Sessions Attempted. The benchmark for Session Establishment Rate is measured with a fixed value for maximum Session Attempts.

Finally, the overall value of these tests is to serve as a comparison function between multiple SIP implementations. One way to use these tests is to derive benchmarks with SIP devices from Vendor-A, derive a new set of benchmarks with similar SIP devices from Vendor-B and perform a comparison on the results of Vendor-A and Vendor-B. This document does not make any claims on the interpretation of such results.

3. Benchmarking Topologies

Familiarity with the benchmarking models in Section 2.2 of [I-D.sip-bench-term] is assumed. Figures 1 through 10 in [I-D.sip-bench-term] contain the canonical topologies that can be used to perform the benchmarking tests listed in this document.

4. Test Setup Parameters

4.1. Selection of SIP Transport Protocol

Test cases may be performed with any transport protocol supported by SIP. This includes, but is not limited to, SIP TCP, SIP UDP, and TLS. The protocol used for the SIP transport protocol must be reported with benchmarking results.

4.2. Signaling Server

The Signaling Server is defined in the companion terminology document, ([I-D.sip-bench-term], Section 3.2.2) It is a SIP-speaking device that complies with RFC 3261. Conformance to [RFC3261] is assumed for all tests. The Signaling Server may be the DUT or a component of a SUT. The Signaling Server may include Firewall and/or NAT functionality. The components of the SUT may be a single physical device or separate devices.

4.3. Associated Media

Some tests require Associated Media to be present for each SIP session. The test topologies to be used when benchmarking SUT performance for Associated Media are shown in [I-D.sip-bench-term],

Figures 4 and 5.

4.4. Selection of Associated Media Protocol

The test cases specified in this document provide SIP performance independent of the protocol used for the media stream. Any media protocol supported by SIP may be used. This includes, but is not limited to, RTP, RTSP, and SRTP. The protocol used for Associated Media MUST be reported with benchmarking results.

4.5. Number of Associated Media Streams per SIP Session

Benchmarking results may vary with the number of media streams per SIP session. When benchmarking a SUT for voice, a single media stream is used. When benchmarking a SUT for voice and video, two media streams are used. The number of Associated Media Streams MUST be reported with benchmarking results.

4.6. Session Duration

SUT performance benchmarks may vary with the duration of SIP sessions. Session Duration MUST be reported with benchmarking results. A Session Duration of zero seconds indicates transmission of a BYE immediately following successful SIP establishment indicate by receipt of a 200 OK. An infinite Session Duration indicates that a BYE is never transmitted.

4.7. Attempted Sessions per Second

DUT and SUT performance benchmarks may vary with the the rate of attempted sessions offered by the Tester. Attempted Sessions per Second MUST be reported with benchmarking results.

4.8. Stress Testing

The purpose of this document is to benchmark SIP performance; this document does not benchmark stability of SIP systems under stressful conditions such as a high rate of Attempted Sessions per Second.

4.9. Benchmarking algorithm

In order to benchmark the test cases uniformly in Section 6, the algorithm described in this section should be used. Both, a prosaic description of the algorithm and a pseudo-code description are provided.

The goal is to find the largest value of a SIP session-request-rate, measured in sessions-per-second, which the DUT/SUT can process with

zero errors. To discover that number, an iterative process (defined below) is used to find a candidate for this rate. Once the candidate rate has been found, the DUT/SUT is subjected to an offered load whose arrival rate is set to that of the candidate rate. This test is run for an extended period of time, which is referred to as infinity, and which is, itself, a parameter of the test labeled T in the pseudo-code. This latter phase of testing is called the steady-state phase. If errors are encountered during this steady-state phase, then the candidate rate is reduced by a defined percent, also a parameter of test, and the steady-state phase is entered again until a final (new) steady-state rate is achieved.

The iterative process itself is defined as follows: a starting rate of 100 sessions per second (sps) is selected. The test is executed for the time period identified by t in the pseudo-code below. If no failures occur, the rate is increased to 150 sps and again tested for time period t. The attempt rate is continuously ramped up until a failure is encountered before the end of the test time t. Then an attempt rate is calculated that is higher than the last successful attempt rate by a quantity equal to half the difference between the rate at which failures occurred and the last successful rate. If this new attempt rate also results in errors, a new attempt rate is tried that is higher than the last successful attempt rate by a quantity equal to half the difference between the rate at which failures occurred and the last successful rate. Continuing in this way, an attempt rate without errors is found. The operator can specify margin of error using the parameter G, measured in units of sessions per second.

The pseudo-code corresponding to the description above follows.

```
; ---- Parameters of test, adjust as needed
t := 5000      ; local maximum; used to figure out largest
               ; value
T := 50000     ; global maximum; once largest value has been
               ; figured out, pump this many requests before calling
               ; the test a success
m := {...}    ; other attributes that affect testing, such
               ; as media streams, etc.
s := 100      ; Initial session attempt rate (in sessions/sec)
G := 5        ; granularity of results - the margin of error in sps
C := 0.05     ; calibration amount: How much to back down if we
               ; have found candidate s but cannot send at rate s for
               ; time T without failures

; ---- End of parameters of test
; ---- Initialization of flags, candidate values and upper bounds
```



```

f := false ; indicates that you had a success after the upper limit
F := false ; indicates that test is done
c := 0      ; indicates that we have found an upper limit

proc main
  find_largest_value ; First, figure out the largest value.

  ; Now that the largest value (saved in s) has been figured out,
  ; use it for sending out s requests/s and send out T requests.

  do {
    send_traffic(s, m, T) ; send_traffic not shown
    if (all requests succeeded) {
      F := true ; test is done
    } else if (one or more requests fail) {
      s := s - (C * s) ; Reduce s by calibration amount
      steady_state
    }
  } while (F == false)
end proc

proc find_largest_value
  ; Iterative process to figure out the largest value we can
  ; handle with no failures
  do {
    send_traffic(s, m, t) ; Send s request/sec with m
                          ; characteristics until t requests have
                          ; been sent
    if (all requests succeeded) {
      s' := s ; save candidate value of metric

      if ( c == 0 ) {
        s := s + (0.5 * s)

      } else if ((c == 1) && (s''-s')) > 2*G ) {
        s := s + ( 0.5 * (s'' - s) );

      } else if ((c == 1) && ((s''-s') <= 2*G ) {
        f := true;

      }
    } else if (one or more requests fail) {
      c := 1 ; we have found an upper bound for the metric
      s'' := s ; save new upper bound
      s := s - (0.5 * (s - s'))
    }
  } while (f == false)
end proc

```

5. Reporting Format

5.1. Test Setup Report

SIP Transport Protocol = _____
(valid values: TCP|UDP|TLS|SCTP|specify-other)
Session Attempt Rate = _____
(session attempts/sec)
IS Media Attempt Rate = _____
(IS media attempts/sec)
Total Sessions Attempted = _____
(total sessions to be created over duration of test)
Media Streams Per Session = _____
(number of streams per session)
Associated Media Protocol = _____
(RTP|RTSP|specify-other)
Media Packet Size = _____
(bytes)
Media Offered Load = _____
(packets per second)
Media Session Hold Time = _____
(seconds)
Establishment Threshold time = _____
(seconds)
Loop Detecting Option = _____
(on|off)
Forking Option
 Number of endpoints request sent to = _____
 (1, means forking is not enabled)
 Type of forking = _____
 (serial|parallel)
Authentication option = _____
 (on|off; if on, please see Notes 2 and 3 below).

Note 1: Total Sessions Attempted is used in the calculation of the Session Establishment Performance ([I-D.sip-bench-term], Section 3.4.5). It is the number of session attempts ([I-D.sip-bench-term], Section 3.1.6) that will be made over the duration of the test.

Note 2: When the Authentication Option is "on" the test tool must be set to ignore 401 and 407 failure responses in any test described as a "test to failure." If this is not done, all such tests will yield trivial benchmarks, as all attempt rates will lead to a failure after the first attempt.

Note 3: When the Authentication Option is "on" the DUT/SUT uses two

transactions instead of one when it is establishing a session or accomplishing a registration. The first transaction ends with the 401 or 407. The second ends with the 200 OK or another failure message. The Test Organization interested in knowing how many times the EA was intended to send a REGISTER as distinct from how many times the EA wound up actually sending a REGISTER may wish to record the following data as well:

Number of responses of the following type:

401:	_____	(if authentication turned on; N/A otherwise)
407:	_____	(if authentication turned on; N/A otherwise)

5.2. Device Benchmarks for IS

Registration Rate = _____
(registrations per second)
Re-registration Rate = _____
(registrations per second)
Session Capacity = _____
(sessions)
Session Overload Capacity = _____
(sessions)
Session Establishment Rate = _____
(sessions per second)
Session Establishment Performance = _____
(total established sessions/total sessions attempted)(no units)
Session Attempt Delay = _____
(seconds)

5.3. Device Benchmarks for NS

IM Rate = _____ (IM messages per second)

6. Test Cases

6.1. Baseline Session Establishment Rate of the test bed

Objective:

To benchmark the Session Establishment Rate of the Emulated Agent (EA) with zero failures.

Procedure:

1. Configure the DUT in the test topology shown in Figure 1 in [I-D.sip-bench-term].
2. Set media streams per session to 0.
3. Execute benchmarking algorithm as defined in Section 4.9 to get the baseline session establishment rate. This rate **MUST** be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: This is the scenario to obtain the maximum Session Establishment Rate of the EA and the test bed when no DUT/SUT is present. The results of this test might be used to normalize test results performed on different test beds or simply to better understand the impact of the DUT/SUT on the test bed in question.

6.2. Session Establishment Rate without media

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with no associated media and zero failures.

Procedure:

1. If the DUT/SUT is being benchmarked as a user agent client or a user agent server, configure the DUT in the test topology shown in Figure 1 or Figure 2 in [I-D.sip-bench-term]. Alternatively, if the DUT is being benchmarked as a proxy or a B2BUA, configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 7 in [I-D.sip-bench-term].
3. Set media streams per session to 0.
4. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate. This rate **MUST** be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: This is the scenario to obtain the maximum Session Establishment Rate of the DUT/SUT.

6.3. Session Establishment Rate with Media not on DUT/SUT

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when Associated Media is included in the benchmark test but the media is not running through the DUT/SUT.

Procedure:

1. If the DUT is being benchmarked as proxy or B2BUA, configure the DUT in the test topology shown in Figure 7 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 in [I-D.sip-bench-term].
3. Set media streams per session to 1.
4. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with media. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with Associated Media with any number of media streams per SIP session are expected to be identical to the Session Establishment Rate results obtained without media in the case where the server is running on a platform separate from the platform on which the Media Relay, NAT or Firewall is running.

6.4. Session Establishment Rate with Media on DUT/SUT

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when Associated Media is included in the benchmark test and the media is running through the DUT/SUT.

Procedure:

1. If the DUT is being benchmarked as a user agent client or a user agent server, configure the DUT in the test topology shown in Figure 3 or Figure 4 of [I-D.sip-bench-term]. Alternatively, if the DUT is being benchmarked as a B2BUA, configure the DUT in the test topology shown in Figure 6 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 9 in [I-D.sip-bench-term].
3. Set media streams per session to 1.
4. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with media. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with Associated Media may be lower than those obtained without media in the case where the server and the NAT, Firewall or Media Relay are running on the same platform.

6.5. Session Establishment Rate with Loop Detection Enabled

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when the Loop Detection option is enabled and no media streams are present.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, and loop detection is supported in the DUT, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term]. If the DUT does not support loop detection, then this step can be skipped.
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0.
4. Turn on the Loop Detection option in the DUT or SUT.
5. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with loop detection enabled. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with Loop Detection may be lower than those obtained without Loop Detection enabled.

6.6. Session Establishment Rate with Forking

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when the Forking Option is enabled.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, and forking is supported in the DUT, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term]. If the DUT does not support forking, then this step can be skipped.
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].

3. Set media streams per session to 0.
4. Set the number of endpoints that will receive the forked invitation to a value of 2 or more (subsequent tests may increase this value at the discretion of the tester.)
5. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with forking. This rate **MUST** be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with Forking may be lower than those obtained without Forking enabled.

6.7. Session Establishment Rate with Forking and Loop Detection

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when both the Forking and Loop Detection Options are enabled.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0.
4. Enable the Loop Detection Options on the DUT.
5. Set the number of endpoints that will receive the forked invitation to a value of 2 or more (subsequent tests may increase this value at the discretion of the tester.)
6. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with forking and loop detection. This rate **MUST** be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with Forking and Loop Detection may be lower than those obtained with only Forking or Loop Detection enabled.

6.8. Session Establishment Rate with TLS Encrypted SIP

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when using TLS encrypted SIP signaling.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0 (media is not used in this test).
4. Configure Tester to enable TLS over the transport being benchmarked. Make a note the transport when compiling results. May need to run for each transport of interest.
5. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with encryption. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with TLS Encrypted SIP may be lower than those obtained with plaintext SIP.

6.9. Session Establishment Rate with IPsec Encrypted SIP

Objective:

To benchmark the Session Establishment Rate of the DUT/SUT with zero failures when using IPsec Encrypted SIP signaling.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0 (media is not used in this test).
4. Configure Tester for IPSec.
5. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with encryption. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with IPSec Encrypted SIP may be lower than those obtained with plaintext SIP.

6.10. Session Establishment Rate with SIP Flooding

Objective:

To benchmark the Session Establishment Rate of the SUT with zero failures when SIP Flooding is occurring.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0.
4. Set s to a high value (e.g., 500) (c.f. Section 4.9).
5. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with flooding. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Establishment Rate results obtained with SIP Flooding may be degraded.

6.11. Maximum Registration Rate

Objective:

To benchmark the maximum registration rate of the DUT/SUT with zero failures.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. Set media streams per session to 0.
4. Set the registration timeout value to at least 3600 seconds.
5. Execute benchmarking algorithm as defined in Section 4.9 to get the maximum registration rate. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results:

6.12. Maximum Re-Registration Rate

Objective:

To benchmark the maximum re-registration rate of the DUT/SUT with zero failures.

Procedure:

1. If the DUT is being benchmarked as a proxy or B2BUA, then configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 8 of [I-D.sip-bench-term].
3. First, execute test detailed in Section 6.11 to register the endpoints with the registrar.
4. After at least 5 minutes of Step 2, but no more than 10 minutes after Step 2 has been performed, execute test detailed in Section 6.11 again (this will count as a re-registration).
5. Execute benchmarking algorithm as defined in Section 4.9 to get the maximum re-registration rate. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: The rate should be at least equal to but not more than the result of Section 6.11.

6.13. Maximum IM Rate**Objective:**

To benchmark the maximum IM rate of the SUT with zero failures.

Procedure:

1. If the DUT/SUT is being benchmarked as a user agent client or a user agent server, configure the DUT in the test topology shown in Figure 1 or Figure 2 in [I-D.sip-bench-term]. Alternatively, if the DUT is being benchmarked as a proxy or a B2BUA, configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 5 in [I-D.sip-bench-term].
3. Execute benchmarking algorithm as defined in Section 4.9 to get the maximum IM rate. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results:

6.14. Session Capacity without Media

Objective:

To benchmark the Session Capacity of the SUT without Associated Media.

Procedure:

1. If the DUT/SUT is being benchmarked as a user agent client or a user agent server, configure the DUT in the test topology shown in Figure 1 or Figure 2 in [I-D.sip-bench-term]. Alternatively, if the DUT is being benchmarked as a proxy or a B2BUA, configure the DUT in the test topology shown in Figure 5 in [I-D.sip-bench-term].
2. Configure a SUT according to the test topology shown in Figure 7 in [I-D.sip-bench-term].
3. Set the media streams per session to be 0.
4. Set the Session Duration to be a value greater than T.
5. Execute benchmarking algorithm as defined in Section 4.9 to get the baseline session establishment rate. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.
6. The Session Capacity is the product of T and the Session Establishment Rate.

Expected Results: The maximum rate at which the DUT/SUT can handle session establishment requests with no media for an infinitely long period with no errors. This is the SIP "throughput" of the system with no media.

6.15. Session Capacity with Media

Objective:

To benchmark the session capacity of the DUT/SUT with Associated Media.

Procedure:

1. Configure the DUT in the test topology shown in Figure 3 or Figure 4 of [I-D.sip-bench-term] depending on whether the DUT is being benchmarked as a user agent client or user agent server. Alternatively, configure the DUT in the test topology shown in Figure 6 or Figure 7 in [I-D.sip-bench-term] depending on whether the DUT is being benchmarked as a B2BUA or as a proxy. If a SUT is being benchmarked, configure the SUT as shown in Figure 9 of [I-D.sip-bench-term].
2. Set the media streams per session to 1.
3. Set the Session Duration to be a value greater than T.
4. Execute benchmarking algorithm as defined in Section 4.9 to get the baseline session establishment rate. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.
5. The Session Capacity is the product of T and the Session Establishment Rate.

Expected Results: Session Capacity results obtained with Associated Media with any number of media streams per SIP session will be less than the Session Capacity results obtained without media.

6.16. Session Capacity with Media and a Media Relay/NAT and/or Firewall

Objective:

To benchmark the Session Establishment Rate of the SUT with Associated Media.

Procedure:

1. Configure the SUT as shown in Figure 7 or Figure 10 in [I-D.sip-bench-term].
2. Set media streams per session to 1.
3. Execute benchmarking algorithm as defined in Section 4.9 to get the session establishment rate with media. This rate MUST be recorded using any pertinent parameters as shown in the reporting format of Section 5.1.

Expected Results: Session Capacity results obtained with Associated Media with any number of media streams per SIP session may be lower than the Session Capacity without Media result if the Media Relay, NAT or Firewall is sharing a platform with the server.

7. IANA Considerations

This document does not requires any IANA considerations.

8. Security Considerations

Documents of this type do not directly affect the security of Internet or corporate networks as long as benchmarking is not performed on devices or systems connected to production networks. Security threats and how to counter these in SIP and the media layer is discussed in RFC3261, RFC3550, and RFC3711 and various other drafts. This document attempts to formalize a set of common methodology for benchmarking performance of SIP devices in a lab environment.

9. Acknowledgments

The authors would like to thank Keith Drage and Daryl Malas for their contributions to this document. Dale Worley provided an extensive review that lead to improvements in the documents. We are grateful to Barry Constantine for providing valuable comments during the document's WGLC.

10. References

10.1. Normative References

- [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.
- [I-D.sip-bench-term] Davids, C., Gurbani, V., and S. Poretsky, "SIP Performance Benchmarking Terminology", draft-ietf-bmwg-sip-bench-term-08 (work in progress), January 2013.

10.2. Informative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

Authors' Addresses

Carol Davids
Illinois Institute of Technology
201 East Loop Road
Wheaton, IL 60187
USA

Phone: +1 630 682 6024
Email: davids@iit.edu

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
1960 Lucent Lane
Rm 9C-533
Naperville, IL 60566
USA

Phone: +1 630 224 0216
Email: vkg@bell-labs.com

Scott Poretsky
Allot Communications
300 TradeCenter, Suite 4680
Woburn, MA 08101
USA

Phone: +1 508 309 2179
Email: sporetsky@allot.com

Benchmarking Methodology Working
Group
Internet-Draft
Intended status: Informational
Expires: July 12, 2013

C. Davids
Illinois Institute of Technology
V. Gurbani
Bell Laboratories, Alcatel-Lucent
S. Poretsky
Allot Communications
January 8, 2013

Terminology for Benchmarking Session Initiation Protocol (SIP)
Networking Devices
draft-ietf-bmwg-sip-bench-term-08

Abstract

This document provides a terminology for benchmarking the SIP performance of networking devices. The term performance in this context means the capacity of the device- or system-under-test to process SIP messages. Terms are included for test components, test setup parameters, and performance benchmark metrics for black-box benchmarking of SIP networking devices. The performance benchmark metrics are obtained for the SIP signaling plane only. The terms are intended for use in a companion methodology document for characterizing the performance of a SIP networking device under a variety of conditions. The intent of the two documents is to enable a comparison of the capacity of SIP networking devices. Test setup parameters and a methodology document are necessary because SIP allows a wide range of configuration and operational conditions that can influence performance benchmark measurements. A standard terminology and methodology will ensure that benchmarks have consistent definition and were obtained following the same procedures.

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 July 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminology	5
2. Introduction	6
2.1. Scope	7
2.2. Benchmarking Models	9
3. Term Definitions	14
3.1. Protocol Components	14
3.1.1. Session	14
3.1.2. Signaling Plane	17
3.1.3. Media Plane	18
3.1.4. Associated Media	18
3.1.5. Overload	19
3.1.6. Session Attempt	20
3.1.7. Established Session	20
3.1.8. Invite-initiated Session (IS)	21
3.1.9. Non-INVITE-initiated Session (NS)	22
3.1.10. Session Attempt Failure	22
3.1.11. Standing Sessions Count	23
3.2. Test Components	23
3.2.1. Emulated Agent	24
3.2.2. Signaling Server	24
3.2.3. SIP-Aware Stateful Firewall	24
3.2.4. SIP Transport Protocol	25
3.3. Test Setup Parameters	26
3.3.1. Session Attempt Rate	26
3.3.2. IS Media Attempt Rate	26
3.3.3. Establishment Threshold Time	27
3.3.4. Session Duration	27
3.3.5. Media Packet Size	28
3.3.6. Media Offered Load	28
3.3.7. Media Session Hold Time	29
3.3.8. Loop Detection Option	29
3.3.9. Forking Option	30
3.4. Benchmarks	31
3.4.1. Registration Rate	31
3.4.2. Session Establishment Rate	31
3.4.3. Session Capacity	32
3.4.4. Session Overload Capacity	33
3.4.5. Session Establishment Performance	33
3.4.6. Session Attempt Delay	34
3.4.7. IM Rate	34
4. IANA Considerations	35
5. Security Considerations	35
6. Acknowledgments	35
7. References	36
7.1. Normative References	36
7.2. Informational References	36

Appendix A. White Box Benchmarking Terminology 37

Authors' Addresses 37

1. Terminology

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 BCP 14, RFC2119 [RFC2119]. RFC 2119 defines the use of these key words to help make the intent of standards track documents as clear as possible. While this document uses these keywords, this document is not a standards track document. The term Throughput is defined in RFC2544 [RFC2544].

For the sake of clarity and continuity, this document adopts the template for definitions set out in Section 2 of RFC 1242 [RFC1242].

The terms Device Under Test (DUT) and System Under Test (SUT) are defined in the following BMWG documents:

Device Under Test (DUT) (c.f., Section 3.1.1 RFC 2285 [RFC2285]).
System Under Test (SUT) (c.f., Section 3.1.2, RFC 2285 [RFC2285]).

Many commonly used SIP terms in this document are defined in RFC 3261 [RFC3261]. For convenience the most important of these are reproduced below. Use of these terms in this document is consistent with their corresponding definition in [RFC3261].

- o Call Stateful: A proxy is call stateful if it retains state for a dialog from the initiating INVITE to the terminating BYE request. A call stateful proxy is always transaction stateful, but the converse is not necessarily true.
- o Stateful Proxy: A logical entity that maintains the client and server transaction state machines defined by this specification during the processing of a request, also known as a transaction stateful proxy. The behavior of a stateful proxy is further defined in Section 16 of RFC 3261 [RFC3261]. A transaction stateful proxy is not the same as a call stateful proxy.
- o Stateless Proxy: A logical entity that does not maintain the client or server transaction state machines defined in this specification when it processes requests. A stateless proxy forwards every request it receives downstream and every response it receives upstream.
- o Back-to-back User Agent: A back-to-back user agent (B2BUA) is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests. Unlike a proxy server, it maintains dialog state and must participate in all requests sent on the dialogues it has established. Since it is a concatenation of a UAC and a UAS, no explicit definitions are needed for its behavior.

- o Loop: A request that arrives at a proxy, is forwarded, and later arrives back at the same proxy. When it arrives the second time, its Request-URI is identical to the first time, and other header fields that affect proxy operation are unchanged, so that the proxy will make the same processing decision on the request it made the first time. Looped requests are errors, and the procedures for detecting them and handling them are described by the SIP protocol[RFC3261] and also by RFC 5393

2. Introduction

Service Providers and IT Organizations deliver Voice Over IP (VoIP) and Multimedia network services based on the IETF Session Initiation Protocol (SIP) [RFC3261]. SIP is a signaling protocol originally intended to be used to dynamically establish, disconnect and modify streams of media between end users. As it has evolved it has been adopted for use in a growing number of services and applications. Many of these result in the creation of a media session, but some do not. Examples of this latter group include text messaging and subscription services. The set of benchmarking terms provided in this document is intended for use with any SIP-enabled device performing SIP functions in the interior of the network, whether or not these result in the creation of media sessions. The performance of end-user devices is outside the scope of this document.

A number of networking devices have been developed to support SIP-based VoIP services. These include SIP Servers, Session Border Controllers (SBC), Back-to-back User Agents (B2BUA), and SIP-Aware Stateful Firewalls. These devices contain a mix of voice and IP functions whose performance may be reported using metrics defined by the equipment manufacturer or vendor. The Service Provider or IT Organization seeking to compare the performance of such devices will not be able to do so using these vendor-specific metrics, whose conditions of test and algorithms for collection are often unspecified. SIP functional elements and the devices that include them can be configured many different ways and can be organized into various topologies. These configuration and topological choices impact the value of any chosen signaling benchmark. Unless these conditions-of-test are defined, a true comparison of performance metrics will not be possible. Some SIP-enabled network devices terminate or relay media as well as signaling. The processing of media by the device impacts the signaling performance. As a result, the conditions-of-test must include information as to whether or not the device under test processes media and if the device does process media, a description of the media handled and the manner in which it is handled. This document and its companion methodology document [I-D.ietf-bmwg-sip-bench-meth] provide a set of black-box benchmarks

for describing and comparing the performance of devices that incorporate the SIP User Agent Client and Server functions and that operate in the network's core.

The definition of SIP performance benchmarks necessarily includes definitions of Test Setup Parameters and a test methodology. These enable the Tester to perform benchmarking tests on different devices and to achieve comparable results. This document provides a common set of definitions for Test Components, Test Setup Parameters, and Benchmarks. All the benchmarks defined are black-box measurements of the SIP signaling plane. The Test Setup Parameters and Benchmarks defined in this document are intended for use with the companion Methodology document. Benchmarks of internal DUT characteristics (also known as white-box benchmarks) such as Session Attempt Arrival Rate, which is measured at the DUT, are described in Appendix A to allow additional characterization of DUT behavior with different distribution models.

2.1. Scope

The scope of this work item is summarized as follows:

- o This terminology document describes SIP signaling performance benchmarks for black-box measurements of SIP networking devices. Stress and debug scenarios are not addressed in this work item.
- o The DUT must be an RFC 3261 capable network equipment. This may be a Registrar, Redirect Server, Stateless Proxy or Stateful Proxy. A DUT MAY also include a B2BUA, SBC functionality. The DUT MAY be a multi-port SIP-to-switched network gateway implemented as a SIP UAC or UAS.
- o The DUT MAY include an internal SIP Application Level Gateway (ALG), firewall, and/or a Network Address Translator (NAT). This is referred to as the "SIP Aware Stateful Firewall."
- o The DUT or SUT MUST NOT be end user equipment, such as personal digital assistant, a computer-based client, or a user terminal.
- o The Tester acts as multiple "Emulated Agents" (EA) that initiate (or respond to) SIP messages as session endpoints and source (or receive) associated media for established connections.
- o SIP Signaling in presence of Media
 - * The media performance is not benchmarked in this work item.
 - * It is RECOMMENDED that SIP signaling plane benchmarks be performed with media present, but this is optional.
 - * The SIP INVITE requests MUST include the SDP body.
 - * The type of DUT dictates whether the associated media streams traverse the DUT or SUT. Both scenarios are within the scope of this work item.
 - * SIP is frequently used to create media streams; the signaling plane and media plane are treated as orthogonal to each other in this document. While many devices support the creation of

media streams, benchmarks that measure the performance of these streams are outside the scope of this document and its companion methodology document [I-D.ietf-bmwg-sip-bench-meth]. Tests may be performed with or without the creation of media streams. The presence or absence of media streams MUST be noted as a condition of the test as the performance of SIP devices may vary accordingly. Even if the media is used during benchmarking, only the SIP performance will be benchmarked, not the media performance or quality.

- o Both INVITE and non-INVITE scenarios (such as Instant Messages or IM) are addressed in this document. However, benchmarking SIP presence is not a part of this work item.
- o Different transport mechanisms -- such as UDP, TCP, SCTP, or TLS -- may be used. The specific transport mechanism MUST be noted as a condition of the test as the performance of SIP devices may vary accordingly.
- o Looping and forking options are also considered since they impact processing at SIP proxies.
- o REGISTER and INVITE requests may be challenged or remain unchallenged for authentication purpose. Whether or not the REGISTER and INVITE requests are challenged is a condition of test which will be recorded along with other such parameters which may impact the SIP performance of the device or system under test.
- o Re-INVITE requests are not considered in scope of this work item since the benchmarks for INVITES are based on the dialog created by the INVITE and not on the transactions that take place within that dialog.
- o Only session establishment is considered for the performance benchmarks. Session disconnect is not considered in the scope of this work item. This is because our goal is to determine the maximum capacity of the device or system under test, that is the number of simultaneous SIP sessions that the device or system can support. It is true that there are BYE requests being created during the test process. These transactions do contribute to the load on the device or system under test and thus are accounted for in the metric we derive. We do not seek a separate metric for the number of BYE transactions a device or system can support.
- o SIP Overload [RFC6357] is within the scope of this work item. We test to failure and then can continue to observe and record the behavior of the system after failures are recorded. The cause of failure is not within the scope of this work. We note the failure and may continue to test until a different failure or condition is encountered. Considerations on how to handle overload are deferred to work progressing in the SOC working group [I-D.ietf-soc-overload-control]. Vendors are, of course, free to implement their specific overload control behavior as the expected test outcome if it is different from the IETF recommendations. However, such behavior MUST be documented and interpreted

appropriately across multiple vendor implementations. This will make it more meaningful to compare the performance of different SIP overload implementations.

- o IMS-specific scenarios are not considered, but test cases can be applied with 3GPP-specific SIP signaling and the P-CSCF as a DUT.

2.2. Benchmarking Models

This section shows ten models to be used when benchmarking SIP performance of a networking device. Figure 1 shows the configuration needed to benchmark the tester itself. This model will be used to establish the limitations of the test apparatus.

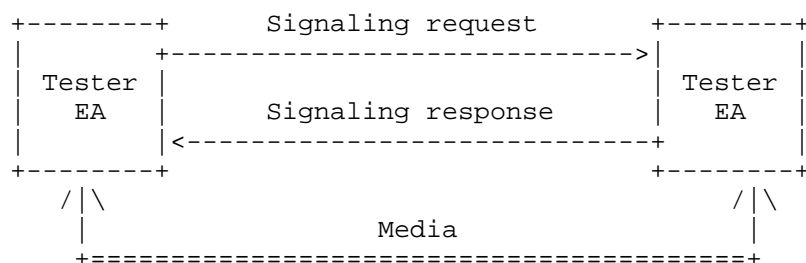


Figure 1: Baseline performance of the Emulated Agent without a DUT present

Figure 2 shows the DUT playing the role of a user agent client (UAC), initiating requests and absorbing responses. This model can be used to baseline the performance of the DUT acting as an UAC without associated media.

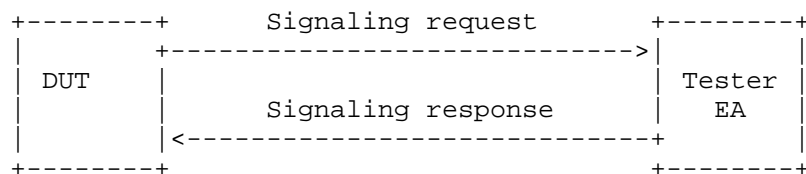


Figure 2: Baseline performance for DUT acting as a user agent client without associated media

Figure 3 shows the DUT playing the role of a user agent server (UAS), absorbing the requests and sending responses. This model can be used as a baseline performance for the DUT acting as a UAS without

associated media.

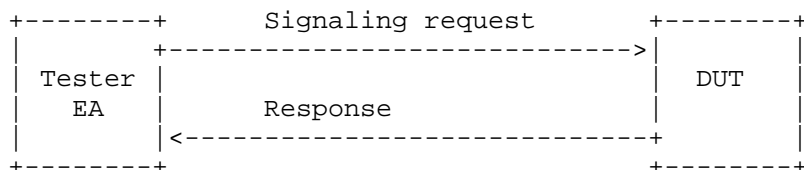


Figure 3: Baseline performance for DUT acting as a user agent server without associated media

Figure 4 shows the DUT plays the role of a user agent client (UAC), initiating requests and absorbing responses. This model can be used as a baseline performance for the DUT acting as a UAC with associated media.

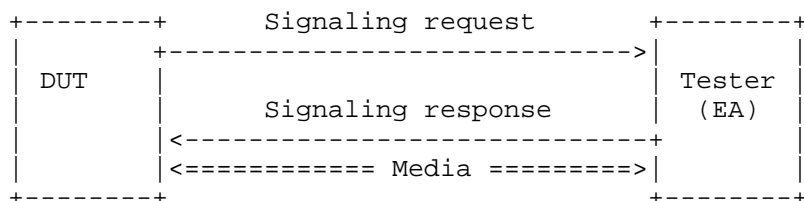


Figure 4: Baseline performance for DUT acting as a user agent client with associated media

Figure 5 shows the DUT plays the role of a user agent server (UAS), absorbing the requests and sending responses. This model can be used as a baseline performance for the DUT acting as a UAS with associated media.

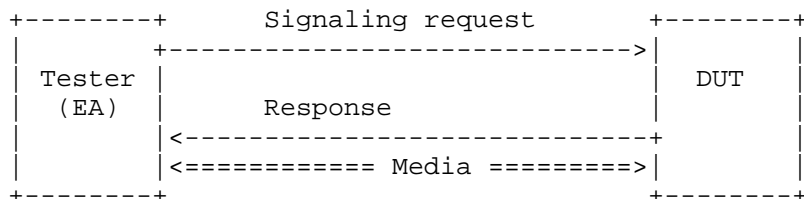


Figure 5: Baseline performance for DUT acting as a user agent server

with associated media

Figure 6 shows that the Tester acts as the initiating and responding EA as the DUT/SUT forwards Session Attempts.

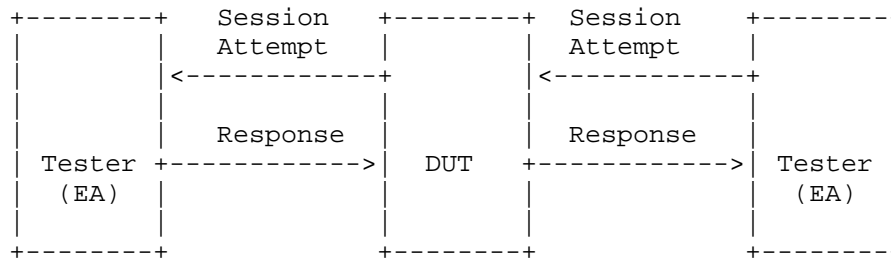


Figure 6: DUT/SUT performance benchmark for session establishment without media

Figure 7 is used when performing those same benchmarks with Associated Media traversing the DUT/SUT.

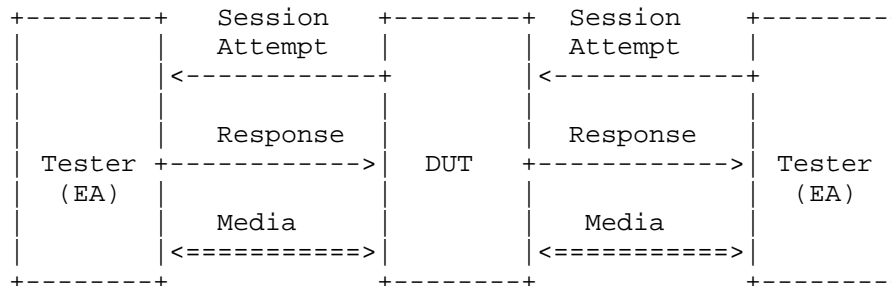


Figure 7: DUT/SUT performance benchmark for session establishment with media traversing the DUT

Figure 8 is to be used when performing those same benchmarks with Associated Media, but the media does not traverse the DUT/SUT. Again, the benchmarking of the media is not within the scope of this work item. The SIP control signaling is benchmarked in the presence of Associated Media to determine if the SDP body of the signaling and the handling of media impacts the performance of the DUT/SUT.

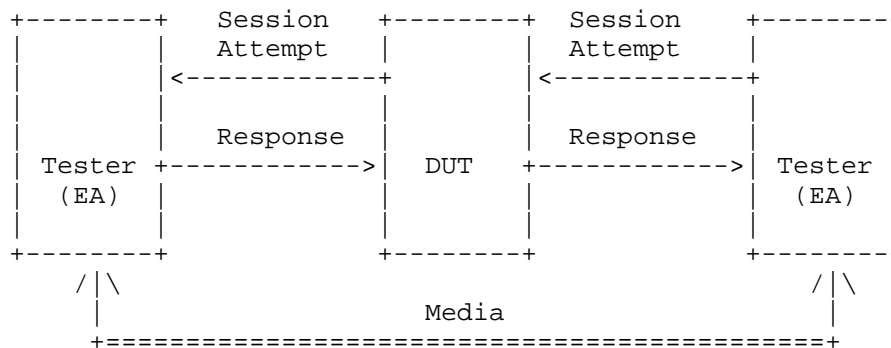


Figure 8: DUT/SUT performance benchmark for session establishment with media external to the DUT

Figure 9 is used when performing benchmarks that require one or more intermediaries to be in the signaling path. The intent is to gather benchmarking statistics with a series of DUTs in place. In this topology, the media is delivered end-to-end and does not traverse the DUT.

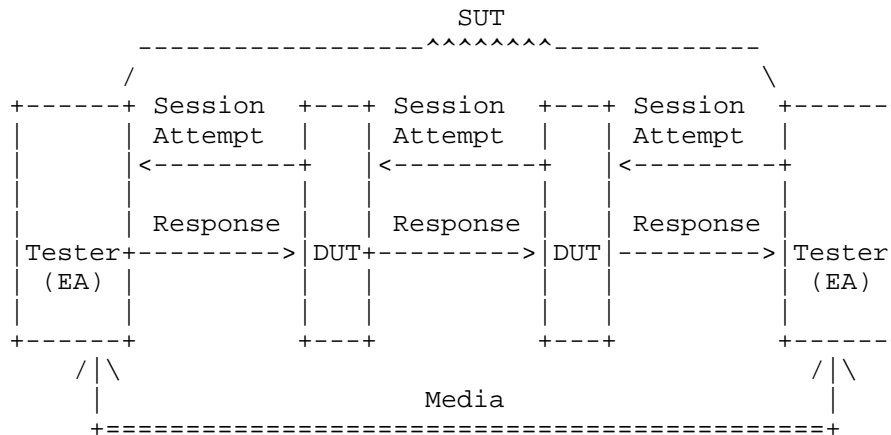


Figure 9: DUT/SUT performance benchmark for session establishment with multiple DUTs and end-to-end media

Figure 10 is used when performing benchmarks that require one or more intermediaries to be in the signaling path. The intent is to gather benchmarking statistics with a series of DUTs in place. In this topology, the media is delivered hop-by-hop through each DUT.

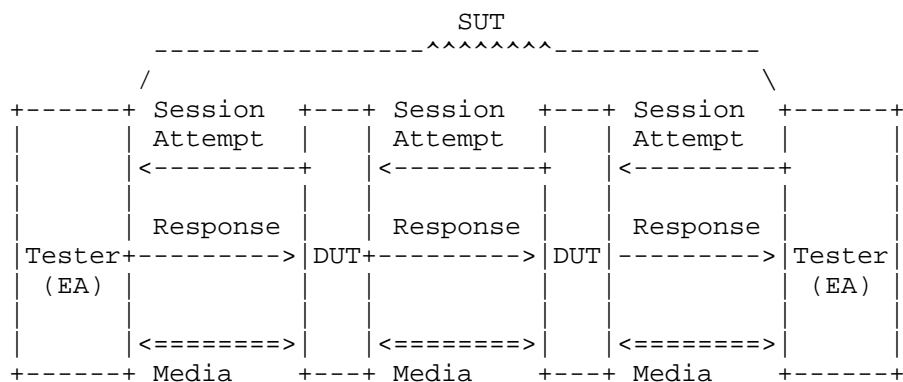


Figure 10: DUT/SUT performance benchmark for session establishment with multiple DUTs and hop-by-hop media

Figure 11 illustrates the SIP signaling for an Established Session. The Tester acts as the EAs and initiates a Session Attempt with the DUT/SUT. When the EA receives a 200 OK from the DUT/SUT that session is considered to be an Established Session. The illustration indicates three states of the session bring created by the EA - (1) Attempting, (2) Established, and (3) Disconnecting. Sessions can be one of two type: Invite-Initiated Session (IS) or Non-Invite Initiated Session (NS). Failure for the DUT/SUT to successfully respond within the Establishment Threshold Time is considered a Session Attempt Failure. SIP Invite messages MUST include the SDP body to specify the Associated Media. Use of Associated Media, to be sourced from the EA, is optional. When Associated Media is used, it may traverse the DUT/SUT depending upon the type of DUT/SUT. The Associated Media is shown in Figure 11 as "Media" connected to media ports M1 and M2 on the EA. After the EA sends a BYE, the session disconnects. Performance test cases for session disconnects are not considered in this work item (the BYE request is shown for completeness.)

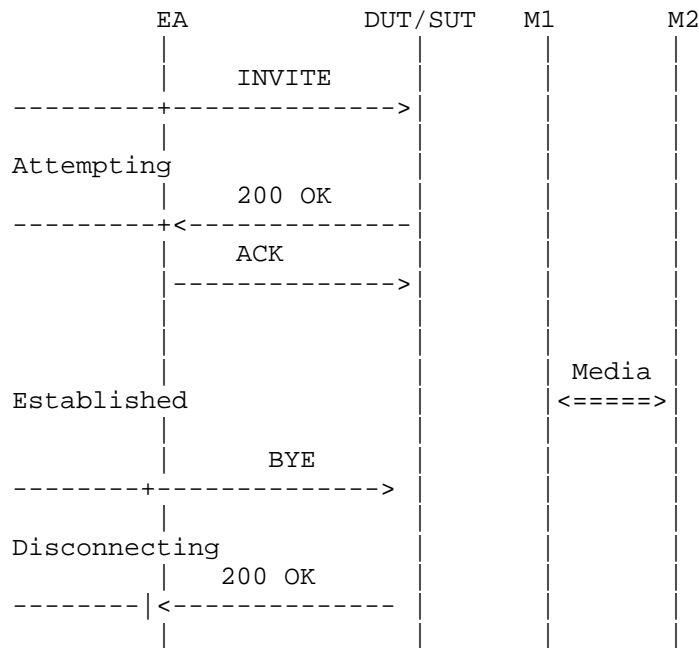


Figure 11: Invite-initiated Session States

3. Term Definitions

3.1. Protocol Components

3.1.1. Session

Definition:

The combination of signaling and media messages and processes that support a SIP-based service.

Discussion:

SIP messages are used to create and manage services for end users. Often, these services include the creation of media streams that are defined in the SDP body of a SIP message and carried in RTP protocol data units. However, SIP messages can also be used to create Instant Message services and subscription services, and such services are not associated with media streams. SIP reserves the term "session" to describe services that are analogous to telephone calls on a circuit switched network. SIP reserves the term "dialog" to refer to a signaling-only relationship between User Agent peers. SIP reserves the term "transaction" to refer to

the brief communication between a client and a server that lasts only until the final response to the SIP request. None of these terms describes the entity whose performance we want to benchmark. For example, the MESSAGE request does not create a dialog and can be sent either within or outside of a dialog. It is not associated with media, but it resembles a phone call in its dependence on human rather than machine initiated responses. The SUBSCRIBE method does create a dialog between the originating end-user and the subscription service. It, too, is not associated with a media session.

In light of the above observations we have extended the term "session" to include SIP-based services that are not initiated by INVITE requests and that do not have associated media. In this extended definition, a session always has a signaling component and may also have a media component. Thus, a session can be defined as signaling-only or a combination of signaling and media. We define the term "Associated Media", see Section 3.1.4, to describe the situation in which media is associated with a SIP dialog. The terminology "Invite-initiated Session" (IS) Section 3.1.8 and "Non-invite-Initiated Session" (NS) Section 3.1.9 are used to distinguish between these two types of session. An Invite-initiated Session is a session as defined in SIP. The performance of a device or system that supports Invite-initiated Sessions that do not create media sessions, "Invite-initiated Sessions without Associated Media", can be measured and is of interest for comparison and as a limiting case. The REGISTER request can be considered to be a "Non-invite-initiated Session without Associated Media." A separate set of benchmarks is provided for REGISTER requests since most implementations of SIP-based services require this request and since a registrar may be a device under test.

A Session in the context of this document, can be considered to be a vector with three components:

1. A component in the signaling plane (SIP messages), sess.sig;
2. A media component in the media plane (RTP and SRTP streams for example), sess.med (which may be null);
3. A control component in the media plane (RTCP messages for example), sess.medc (which may be null).

An IS is expected to have non-null sess.sig and sess.med components. The use of control protocols in the media component is media dependent, thus the expected presence or absence of sess.medc is media dependent and test-case dependent. An NS is expected to have a non-null sess.sig component, but null sess.med and sess.medc components.

Packets in the Signaling Plane and Media Plane will be handled by different processes within the DUT. They will take different paths within a SUT. These different processes and paths may produce variations in performance. The terminology and benchmarks defined in this document and the methodology for their use are designed to enable us to compare performance of the DUT/SUT with reference to the type of SIP-supported application it is handling.

Note that one or more sessions can simultaneously exist between any participants. This can be the case, for example, when the EA sets up both an IM and a voice call through the DUT/SUT. These sessions are represented as an array session[x].

Sessions will be represented as a vector array with three components, as follows:

session->

session[x].sig, the signaling component

session[x].medc[y], the media control component (e.g. RTCP)

session[x].med[y], an array of associated media streams (e.g. RTP, SRTP, RTSP, MSRP). This media component may consist of zero or more media streams.

Figure 12 models the vectors of the session.

Measurement Units:

N/A.

Issues:

None.

See Also:

Media Plane

Signaling Plane

Associated Media

Invite-initiated Session (IS)

Non-invite-initiated Session (NS)

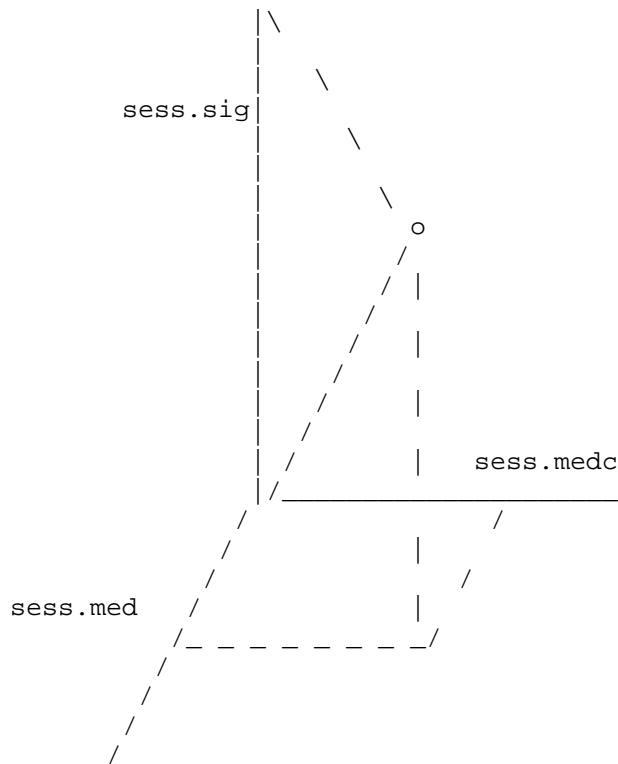


Figure 12: Session components

3.1.2. Signaling Plane

Definition:

The plane in which SIP messages [RFC3261] are exchanged between SIP Agents [RFC3261].

Discussion:

SIP messages are used to establish sessions in several ways: directly between two User Agents [RFC3261], through a Proxy Server [RFC3261], or through a series of Proxy Servers. The Session Description Protocol (SDP) is included in the Signaling Plane. The Signaling Plane for a single Session is represented by session.sig.

Measurement Units:

N/A.

Issues:

None.

See Also:

Media Plane

EAs

3.1.3. Media Plane

Definition:

The data plane in which one or more media streams and their associated media control protocols are exchanged between User Agents after a media connection has been created by the exchange of signaling messages in the Signaling Plane.

Discussion:

Media may also be known as the "bearer channel". The Media Plane MUST include the media control protocol, if one is used, and the media stream(s). Examples of media are audio and video. The media streams are described in the SDP of the Signaling Plane. The media for a single Session is represented by session.med. The media control protocol for a single media description is represented by session.medc.

Measurement Units:

N/A.

Issues:

None.

See Also:

Signaling Plane

3.1.4. Associated Media

Definition:

Media that corresponds to an 'm' line in the SDP payload of the Signaling Plane.

Discussion:

Any media protocol MAY be used.

For any session's signaling component, `session.sig`, there may be zero, one, or multiple associated media streams. When there are multiple media streams, these are represented by a vector array `session.med[y]`. When there are multiple media streams there will be multiple media control protocol descriptions as well. They are represented by a vector array `session.medc[y]`.

Measurement Units:

N/A.

Issues:

None.

3.1.5. Overload

Definition:

Overload is defined as the state where a SIP server does not have sufficient resources to process all incoming SIP messages [RFC6357].

Discussion:

The distinction between an overload condition and other failure scenarios is outside the scope of black box testing and of this document. Under overload conditions, all or a percentage of Session Attempts will fail due to lack of resources. In black box testing the cause of the failure is not explored. The fact that a failure occurred for whatever reason, will trigger the tester to reduce the offered load, as described in the companion methodology document, [I-D.ietf-bmwg-sip-bench-meth]. SIP server resources may include CPU processing capacity, network bandwidth, input/output queues, or disk resources. Any combination of resources may be fully utilized when a SIP server (the DUT/SUT) is in the overload condition. For proxy-only type of devices, it is expected that the proxy will be driven into overload based on the delivery rate of signaling requests.

For UA-type of network devices such as gateways, it is expected that the UA will be driven into overload based on the volume of media streams it is processing.

Measurement Units:

N/A.

Issues:

The issue of overload in SIP networks is currently a topic of discussion in the SIPPING WG. The normal response to an overload stimulus -- sending a 503 response -- is considered inadequate and new response codes and behaviors may be specified in the future. From the perspective of this document, all these responses will be considered to be failures. There is thus no dependency between this document and the ongoing work on the treatment of overload failure.

3.1.6. Session Attempt**Definition:**

A SIP request sent by the EA that has not received a final response.

Discussion:

The attempted session may be Invite Initiated or Non-invite Initiated. When counting the number of session attempts we include all INVITEs that are rejected for lack of authentication information. The EA needs to record the total number of session attempts including those attempts that are routinely rejected by a proxy that requires the UA to authenticate itself. The EA is provisioned to deliver a specific number of session attempts per second. But the EA must also count the actual number of session attempts per given tie interval.

Measurement Units:

N/A.

Issues:

None.

See Also:

Session
Session Attempt Rate
Invite-initiated Session
Non-Invite initiated Session

3.1.7. Established Session**Definition:**

A SIP session for which the EA acting as the UE/UA has received a 200 OK message.

Discussion:

An Established Session MAY be Invite Initiated or Non-invite Initiated.

Measurement Units:

N/A.

Issues:

None.

See Also:

Invite-initiated Session
Session Attempting State
Session Disconnecting State

3.1.8. Invite-initiated Session (IS)**Definition:**

A Session that is created by an exchange of messages in the Signaling Plane, the first of which is a SIP INVITE request.

Discussion:

When an IS becomes an Established Session its signaling component is identified by the SIP dialog parameter values, Call-ID, To-tag, and From-tag (RFC3261 [RFC3261]). An IS may have zero, one or multiple Associated Media descriptions in the SDP body. The inclusion of media is test case dependent. An IS is successfully established if the following two conditions are met:

1. Sess.sig is established by the end of Establishment Threshold Time (c.f. Section 3.3.3), and
2. If a media session is described in the SDP body of the signaling message, then the media session is established by the end of Establishment Threshold Time (c.f. Section 3.3.3). An SBC or B2BUA may receive media from a calling or called party before a signaling dialog is established and certainly before a confirmed dialog is established. The EA can be built in such a way that it does not send early media or it needs to include a parameter that indicates when it will send media. This parameter must be included in the list of test setup parameters in Section 5.1 of [I-D.ietf-bmwg-sip-bench-meth]

Measurement Units:

N/A.

Issues:

None.

See Also:

Session

Non-Invite initiated Session

Associated Media

3.1.9. Non-INVITE-initiated Session (NS)

Definition:

A session that is created by an exchange of SIP messages in the Signaling Plane the first of which is not a SIP INVITE message.

Discussion:

An NS is successfully established if the Session Attempt via a non- INVITE request results in the EA receiving a 2xx reply before the expiration of the Establishment Threshold timer (c.f., Section 3.3.3). An example of a NS is a session created by the SUBSCRIBE request.

Measurement Units:

N/A.

Issues:

None.

See Also:

Session

Invite-initiated Session

3.1.10. Session Attempt Failure

Definition:

A session attempt that does not result in an Established Session.

Discussion:

The session attempt failure may be indicated by the following observations at the EA:

1. Receipt of a SIP 4xx, 5xx, or 6xx class response to a Session Attempt.
2. The lack of any received SIP response to a Session Attempt within the Establishment Threshold Time (c.f. Section 3.3.3).

Measurement Units:

N/A.

Issues:

None.

See Also:

Session Attempt

3.1.11. Standing Sessions Count

Definition:

The number of Sessions currently established on the DUT/SUT at any instant.

Discussion:

The number of Standing Sessions is influenced by the Session Duration and the Session Attempt Rate. Benchmarks MUST be reported with the maximum and average Standing Sessions for the DUT/SUT for the duration of the test. In order to determine the maximum and average Standing Sessions on the DUT/SUT for the duration of the test it is necessary to make periodic measurements of the number of Standing Sessions on the DUT/SUT. The recommended value for the measurement period is 1 second. Since we cannot directly poll the DUT/SUT, we take the number of standing sessions on the DUT/SUT to be the number of distinct calls as measured by the number of distinct Call-IDs that the EA is processing at the time of measurement. The EA must make that count available for viewing and recording.

Measurement Units:

Number of sessions

Issues:

None.

See Also:

Session Duration
Session Attempt Rate
Session Attempt Rate
Emulated Agent

3.2. Test Components

3.2.1. Emulated Agent

Definition:

A device in the test topology that initiates/responds to SIP messages as one or more session endpoints and, wherever applicable, sources/receives Associated Media for Established Sessions.

Discussion:

The EA functions in the Signaling and Media Planes. The Tester may act as multiple EAs.

Measurement Units:

N/A

Issues:

None.

See Also:

Media Plane
Signaling Plane
Established Session
Associated Media

3.2.2. Signaling Server

Definition:

Device in the test topology that acts to create sessions between EAs. This device is either a DUT or a component of a SUT.

Discussion:

The DUT MUST be an RFC 3261 capable network equipment such as a Registrar, Redirect Server, User Agent Server, Stateless Proxy, or Stateful Proxy. A DUT MAY also include B2BUA or SBC.

Measurement Units:

NA

Issues:

None.

See Also:

Signaling Plane

3.2.3. SIP-Aware Stateful Firewall

Definition:

Device in the test topology that provides protection against various types of security threats to which the Signaling and Media Planes of the EAs and Signaling Server are vulnerable.

Discussion:

Threats may include Denial-of-Service, theft of service and misuse of service. The SIP-Aware Stateful Firewall MAY be an internal component or function of the Session Server. The SIP-Aware Stateful Firewall MAY be a standalone device. If it is a standalone device it MUST be paired with a Signaling Server. If it is a standalone device it MUST be benchmarked as part of a SUT. SIP-Aware Stateful Firewalls MAY include Network Address Translation (NAT) functionality. Ideally, the inclusion of the SIP-Aware Stateful Firewall in the SUT does not lower the measured values of the performance benchmarks.

Measurement Units:

N/A

Issues:

None.

See Also:

3.2.4. SIP Transport Protocol

Definition:

The protocol used for transport of the Signaling Plane messages.

Discussion:

Performance benchmarks may vary for the same SIP networking device depending upon whether TCP, UDP, TLS, SCTP, or another transport layer protocol is used. For this reason it MAY be necessary to measure the SIP Performance Benchmarks using these various transport protocols. Performance Benchmarks MUST report the SIP Transport Protocol used to obtain the benchmark results.

Measurement Units:

TCP,UDP, SCTP, TLS over TCP, TLS over UDP, or TLS over SCTP

Issues:

None.

See Also:

3.3. Test Setup Parameters

3.3.1. Session Attempt Rate

Definition:

Configuration of the EA for the number of sessions per second that the EA attempts to establish using the services of the DUT/SUT.

Discussion:

The Session Attempt Rate is the number of sessions per second that the EA sends toward the DUT/SUT. Some of the sessions attempted may not result in a session being established. A session in this case may be either an IS or an NS.

Measurement Units:

Session attempts per second

Issues:

None.

See Also:

Session

Session Attempt

3.3.2. IS Media Attempt Rate

Definition:

Configuration on the EA for the rate, measured in sessions per second, at which the EA attempts to establish INVITE-initiated sessions with Associated Media, using the services of the DUT/SUT.

Discussion:

An IS is not required to include a media description. The IS Media Attempt Rate defines the number of media sessions we are trying to create, not the number of media sessions that are actually created. Some attempts might not result in successful sessions established on the DUT.

Measurement Units:

session attempts per second (saps)

Issues:

None.

See Also:
IS

3.3.3. Establishment Threshold Time

Definition:

Configuration of the EA for representing the amount of time that an EA will wait before declaring a Session Attempt Failure.

Discussion:

This time duration is test dependent.

It is RECOMMENDED that the Establishment Threshold Time value be set to Timer B (for ISs) or Timer F (for NSs) as specified in RFC 3261, Table 4 [RFC3261]. Following the default value of T1 (500ms) specified in the table and a constant multiplier of 64 gives a value of 32 seconds for this timer (i.e., 500ms * 64 = 32s).

Measurement Units:
seconds

Issues:
None.

See Also:
session establishment failure

3.3.4. Session Duration

Definition:

Configuration of the EA that represents the amount of time that the SIP dialog is intended to exist between the two EAs associated with the test.

Discussion:

The time at which the BYE is sent will control the Session Duration

Normally the Session Duration will be the same as the Media Session Hold Time. However, it is possible that the dialog established between the two EAs can support different media sessions at different points in time. Providing both parameters allows the testing agency to explore this possibility.

Measurement Units:
seconds

Issues:
None.

See Also:
Media Session Hold Time

3.3.5. Media Packet Size

Definition:
Configuration on the EA for a fixed size of packets used for media streams.

Discussion:
For a single benchmark test, all sessions use the same size packet for media streams. The size of packets can cause variation in performance benchmark measurements.

Measurement Units:
bytes

Issues:
None.

See Also:

3.3.6. Media Offered Load

Definition:
Configuration of the EA for the constant rate of Associated Media traffic offered by the EA to the DUT/SUT for one or more Established Sessions of type IS.

Discussion:
The Media Offered Load to be used for a test MUST be reported with three components:
1. per Associated Media stream;
2. per IS;
3. aggregate.
For a single benchmark test, all sessions use the same Media Offered Load per Media Stream. There may be multiple Associated Media streams per IS. The aggregate is the sum of all Associated Media for all IS.

Measurement Units:
packets per second (pps)

Issues:
None.

See Also:
Established Session
Invite Initiated Session
Associated Media

3.3.7. Media Session Hold Time

Definition:
Parameter configured at the EA, that represents the amount of time that the Associated Media for an Established Session of type IS will last.

Discussion:
The Associated Media streams may be bi-directional or uni-directional as indicated in the test methodology. Normally the Media Session Hold Time will be the same as the Session Duration. However, it is possible that the dialog established between the two EAs can support different media sessions at different points in time. Providing both parameters allows the testing agency to explore this possibility.

Measurement Units:
seconds

Issues:
None.

See Also:
Associated Media
Established Session
Invite-initiated Session (IS)

3.3.8. Loop Detection Option

Definition:
An option that causes a Proxy to check for loops in the routing of a SIP request before forwarding the request.

Discussion:

This is an optional process that a SIP proxy may employ; the process is described under Proxy Behavior in RFC 3261 [RFC3261] in Section 16.3 Request Validation and that section also contains suggestions as to how the option could be implemented. Any procedure to detect loops will use processor cycles and hence could impact the performance of a proxy.

Measurement Units:

N/A

Issues:

None.

See Also:**3.3.9. Forking Option****Definition:**

An option that enables a Proxy to fork requests to more than one destination.

Discussion:

This is an process that a SIP proxy may employ to find the UAS. The option is described under Proxy Behavior in RFC 3261 in Section 16.1. A proxy that uses forking must maintain state information and this will use processor cycles and memory. Thus the use of this option could impact the performance of a proxy and different implementations could produce different impacts. SIP supports serial or parallel forking. When performing a test, the type of forking mode MUST be indicated.

Measurement Units:

The number of endpoints that will receive the forked invitation. A value of 1 indicates that the request is destined to only one endpoint, a value of 2 indicates that the request is forked to two endpoints, and so on. This is an integer value ranging between 1 and N inclusive, where N is the maximum number of endpoints to which the invitation is sent.
Type of forking used, namely parallel or serial.

Issues:

None.

See Also:

3.4. Benchmarks

3.4.1. Registration Rate

Definition:

The maximum number of registrations that can be successfully completed by the DUT/SUT in a given time period without registration failures in that time period.

Discussion:

This benchmark is obtained with zero failure in which 100% of the registrations attempted by the EA are successfully completed by the DUT/SUT. The registration rate provisioned on the Emulated Agent is raised and lowered as described in the algorithm in the companion methodology draft [I-D.ietf-bmwg-sip-bench-meth] until a traffic load consisting of registrations at the given attempt rate over the sustained period of time identified by T in the algorithm completes without failure.

Measurement Units:

registrations per second (rps)

Issues:

None.

See Also:

3.4.2. Session Establishment Rate

Definition:

The maximum number of sessions that can be successfully completed by the DUT/SUT in a given time period without session establishment failures in that time period.

Discussion:

This benchmark is obtained with zero failure in which 100% of the sessions attempted by the Emulated Agent are successfully completed by the DUT/SUT. The session attempt rate provisioned on the EA is raised and lowered as described in the algorithm in the accompanying methodology document, until a traffic load at the given attempt rate over the sustained period of time identified by T in the algorithm completes without any failed session attempts. Sessions may be IS or NS or a mix of both and will be defined in the particular test.

Measurement Units:
sessions per second (sps)

Issues:
None.

See Also:
Invite-initiated Sessions
Non-INVITE initiated Sessions
Session Attempt Rate

3.4.3. Session Capacity

Definition:
The maximum value of Standing Sessions Count achieved by the DUT/SUT during a time period T in which the EA is sending session establishment messages at the Session Establishment Rate.

Discussion:
Sessions may be IS or NS. If they are IS they can be with or without media. When benchmarking Session Capacity for sessions with media it is required that these sessions be permanently established, i.e., they remain active for the duration of the test. In the signaling plane, this requirement means that the dialog lasts as long as the test lasts. When media is present, the Media Session Hold Time MUST be set to infinity so that sessions remain established for the duration of the test. If the DUT/SUT is dialog-stateful, then we expect its performance will be impacted by setting Media Session Hold Time to infinity, since the DUT/SUT will need to allocate resources to process and store the state information. The report of the Session Capacity must include the Session Establishment Rate at which it was measured.

Measurement Units:
sessions

Issues:
None.

See Also:
Established Session
Session Attempt Rate
Session Attempt Failure

3.4.4. Session Overload Capacity

Definition:

The maximum number of Established Sessions that can exist simultaneously on the DUT/SUT until it stops responding to Session Attempts.

Discussion:

Session Overload Capacity is measured after the Session Capacity is measured. The Session Overload Capacity is greater than or equal to the Session Capacity. When benchmarking Session Overload Capacity, continue to offer Session Attempts to the DUT/SUT after the first Session Attempt Failure occurs and measure Established Sessions until there is no SIP message response for the duration of the Establishment Threshold. Note that the Session Establishment Performance is expected to decrease after the first Session Attempt Failure occurs.

Units:

Sessions

Issues:

None.

See Also:

Overload
Session Capacity
Session Attempt Failure

3.4.5. Session Establishment Performance

Definition:

The percent of Session Attempts that become Established Sessions over the duration of a benchmarking test.

Discussion:

Session Establishment Performance is a benchmark to indicate session establishment success for the duration of a test. The duration for measuring this benchmark is to be specified in the Methodology. The Session Duration SHOULD be configured to infinity so that sessions remain established for the entire test duration.

Session Establishment Performance is calculated as shown in the following equation:

$$\text{Session Establishment Performance} = \frac{\text{Total Established Sessions}}{\text{Total Session Attempts}}$$

Session Establishment Performance may be monitored real-time during a benchmarking test. However, the reporting benchmark MUST be based on the total measurements for the test duration.

Measurement Units:
Percent (%)

Issues:
None.

See Also:
Established Session
Session Attempt

3.4.6. Session Attempt Delay

Definition:
The average time measured at the EA for a Session Attempt to result in an Established Session.

Discussion:
Time is measured from when the EA sends the first INVITE for the call-ID in the case of an IS. Time is measured from when the EA sends the first non-INVITE message in the case of an NS. Session Attempt Delay MUST be measured for every established session to calculate the average. Session Attempt Delay MUST be measured at the Session Establishment Rate.

Measurement Units:
Seconds

Issues:
None.

See Also:
Session Establishment Rate

3.4.7. IM Rate

Definition:
Maximum number of IM messages completed by the DUT/SUT.

Discussion:
For a UAS, the definition of success is the receipt of an IM request and the subsequent sending of a final response.

For a UAC, the definition of success is the sending of an IM request and the receipt of a final response to it. For a proxy, the definition of success is as follows:

- A. the number of IM requests it receives from the upstream client MUST be equal to the number of IM requests it sent to the downstream server; and
- B. the number of IM responses it receives from the downstream server MUST be equal to the number of IM requests sent to the downstream server; and
- C. the number of IM responses it sends to the upstream client MUST be equal to the number of IM requests it received from the upstream client.

Measurement Units:

IM messages per second

Issues:

None.

See Also:

4. IANA Considerations

This document requires no IANA considerations.

5. Security Considerations

Documents of this type do not directly affect the security of Internet or corporate networks as long as benchmarking is not performed on devices or systems connected to production networks. Security threats and how to counter these in SIP and the media layer is discussed in RFC3261 [RFC3261], RFC 3550 [RFC3550], RFC3711 [RFC3711] and various other drafts. This document attempts to formalize a set of common terminology for benchmarking SIP networks. Packets with unintended and/or unauthorized DSCP or IP precedence values may present security issues. Determining the security consequences of such packets is out of scope for this document.

6. Acknowledgments

The authors would like to thank Keith Drage, Cullen Jennings, Daryl Malas, Al Morton, and Henning Schulzrinne for invaluable contributions to this document. Dale Worley provided an extensive review that lead to improvements in the documents. We are grateful to Barry Constantine for providing valuable comments during the

document's WGLC.

7. References

7.1. Normative References

- [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.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [I-D.ietf-bmwg-sip-bench-meth] Davids, C., Gurbani, V., and S. Poretsky, "Methodology for Benchmarking SIP Networking Devices", draft-ietf-bmwg-sip-bench-meth-08 (work in progress), January 2013.

7.2. Informational References

- [RFC2285] Mandeville, R., "Benchmarking Terminology for LAN Switching Devices", RFC 2285, February 1998.
- [RFC1242] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC6357] Hilt, V., Noel, E., Shen, C., and A. Abdelal, "Design Considerations for Session Initiation Protocol (SIP) Overload Control", RFC 6357, August 2011.
- [I-D.ietf-soc-overload-control] Gurbani, V., Hilt, V., and H. Schulzrinne, "Session Initiation Protocol (SIP) Overload Control", draft-ietf-soc-overload-control-11 (work in progress),

November 2012.

Appendix A. White Box Benchmarking Terminology

Session Attempt Arrival Rate

Definition:

The number of Session Attempts received at the DUT/SUT over a specified time period.

Discussion:

Sessions Attempts are indicated by the arrival of SIP INVITES OR SUBSCRIBE NOTIFY messages. Session Attempts Arrival Rate distribution can be any model selected by the user of this document. It is important when comparing benchmarks of different devices that same distribution model was used. Common distributions are expected to be Uniform and Poisson.

Measurement Units:

Session attempts/sec

Issues:

None.

See Also:

Session Attempt

Authors' Addresses

Carol Davids
Illinois Institute of Technology
201 East Loop Road
Wheaton, IL 60187
USA

Phone: +1 630 682 6024
Email: davids@iit.edu

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
1960 Lucent Lane
Rm 9C-533
Naperville, IL 60566
USA

Phone: +1 630 224 0216
Email: vkg@bell-labs.com

Scott Poretsky
Allot Communications
300 TradeCenter, Suite 4680
Woburn, MA 08101
USA

Phone: +1 508 309 2179
Email: sporetsky@allot.com

Benchmarking Methodology Working Group
Internet-Draft
Intended status: Informational
Expires: July 30, 2013

V. Manral
P. Sharma
HP
Y. Ping
H3C
January 26, 2013

Benchmarking Power usage of networking devices
draft-manral-bmwg-power-usage-03

Abstract

With the rapid growth of networks around the globe there is an ever increasing need to improve the energy efficiency of devices. Operators beginning to seek more information of power consumption in the network, have no standard mechanism to measure, report and compare power usage of different networking equipment under different network configuration and conditions exist.

This document provides suggestions for measuring power usage of live networks under different traffic loads and various switch router configuration settings. It provides a suite which can be deployed on any networking device .

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 July 30, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Challenges in defining benchmarks	4
3. Factors for power consumption	5
3.1. Network Factors affecting power consumption	5
3.2. Device Factors affecting power consumption	5
3.3. Traffic Factors affecting power consumption	6
4. Network Energy Consumption Rate (NECR)	7
5. Network Energy Proportionality Index (NEPI)	8
6. Benchmark Details	9
7. Security Considerations	10
8. IANA Considerations	11
9. Acknowledgements	12
10. References	13
10.1. Normative References	13
10.2. Informative References	13
Authors' Addresses	14

1. Introduction

Energy Efficiency is becoming increasingly important in the operation of network infrastructure. Data traffic is exploding at an accelerated rate. Networks provide communication channels that facilitates components of the infrastructures to exchange critical information and are always on. On the other hand, a lot of devices run at very low average utilization rates. Various strategies are being defined to improve network utilization of these devices and thus improve power consumption.

The first step to obtain a network wide view is to start with an individual device view of the system and address different devices in the network on a per device basis. The easiest way to measure the power consumption of a device is to use a power meter. This can be used to measure power under a variety of conditions affecting power usage on a networking device.

Various techniques have been defined for energy management of networking devices. However, there is no common strategy to actually benchmark power utilization of networking devices like routers or switches. This document defines the mechanism to correctly characterize and benchmark the power consumption of various networking devices so as to be able to correctly measure and compare the power usage of various devices. This will enable intelligent decisions to optimize the power consumption for individual devices and the network as a whole. Benchmark are also required to compare effectiveness of various energy optimization techniques.

The Network Energy Consumption Rate (NECR) as well as Network Energy Proportionality Index (NEPI) is also defined here.

The procedures/ metrics defined in this document have been used to perform live measurement with a variety of networking equipment from three large well known vendors.

2. Challenges in defining benchmarks

Using the "Maximum Rated Power" and spec sheets of devices and adding the values for all devices are of little use because the measurement gives the maximum power that can be consumed by the device, however that does not accurately reflect the power consumed by the device under a normal work load. Typical energy requirements of a networking device are dependent on device configuration and traffic.

The ratio of the actual power consumed by the device on an average, to its maximum rated power varies widely across different device families. Thus, relying merely on the maximum rated power can grossly overestimate the total energy consumed by networking equipment.

There are a wide variety of networking equipment and finding a general benchmark to work across a variety of devices, requires a lot of flexibility in benchmarking methodology. The workload and test conditions will also depend on the kind of device.

A network device consists of a lot of individual components, each of which consumes power. For example, only considering the power consumption of the CPU/ data forwarding ASIC we may ignore the power consumption of the other components like external memory.

Power instrumentation of a device in a live network involves unplugging the device and plugging it into a power meter. This can in turn lead to traffic loss. Unfortunately, most current equipment is not equipped with internal instrumentation to report power usage of the device or its components. It is for this reason the power measurement is done on an individual device under different network conditions using a traffic generator.

The network devices can also dissipate significant heat. Past studies have shown dissipation ratios of 2.5. Which means if the power in is 2.5 Watt, only 1 Watt is used for actual work, the rest is dissipated as heat. This heating can lead to more power consumed by fan/ compressor for cooling the devices. Though this methodology does not measure the power consumed by external cooling infrastructure, it measures the power consumed internally. It also (optionally) measures the temperature change of the device which can be correlated to the amount of external power consumed to cool the device.

The amount of power used at startup can be more than the average power usage of the device. This is also measured as part of the test methodology.

3. Factors for power consumption

The metrics defined here will help operators get a more accurate idea of power consumed by network equipment and hence forecast their power budget. These will also help device vendors test and compare the new power efficiency enhancements on various devices.

3.1. Network Factors affecting power consumption

The first and the most important factor from the network perspective which can determine the power consumption is the traffic load. Benchmarks must be performed with different traffic loads in the network.

There are now various kinds of transceivers/ connectors on a network device. For the same bandwidth the power usage of a device depends on the kind of connector used. The connector/ interface type used needs to be specified in the benchmark.

The length of the cable used also defines the amount of power consumed by the system. Benchmarks should specify the cable length used. For example, a 5 meter cable can be used wherever possible.

3.2. Device Factors affecting power consumption

Base Chassis Power - typically, higher end network devices come with a chassis and card slots. Each slot may have a number of ports. For the lower end devices there are no removable card slots. In both these cases the base chassis power consists of processors, fans, memory, etc.

Number of line cards - In switches that support inserting linecards, there is a limit on the number of ports per linecard as well as the aggregate bandwidth that each linecard can accommodate. This mechanism allows network operators the flexibility to only plug in as many linecards as they need. For each benchmark the total number of line cards plugged into the system needs to be specified.

Number of active ports - This term refers to the total number of ports on the switch (across all the linecards) that are active (with cables plugged in). The remaining ports on the switch are explicitly disabled using the switchs command line interface. For each benchmark the number of active and passive ports must be specified.

Port settings - Setting this parameter limits the line rate forwarding capacity of individual ports. For each benchmark the port configuration and settings need to be specified.

Port Utilization - This term describes the actual throughput flowing through a port relative to its specified capacity. For each benchmark the port utilization of each port must be specified. The actual traffic can use the information defined in RFC 2544 [RFC2544].

TCAM - Network vendors typically implement packet classification in hardware. TCAMs are supported by most vendors as they have very fast look-up times. However, they are notoriously power-hungry. The size of the TCAM in a switch is widely variable. The size of the TCAM needs to be reported in the benchmark document. The number of TCAM entries does not affect power consumption.

Firmware - Vendors periodically release upgraded versions of their switch/router firmware. Different versions of firmware may also impact the device power consumption. The firmware version needs to be reported in the benchmark document. Different firmware versions have resulted in different power usage.

3.3. Traffic Factors affecting power consumption

Packet Size - Different packet sizes typically do not effect power consumption.

Inter-Packet Delay - time between successive packets may affect power usage but we do not measure the effects in detail.

CPU traffic - Percentage of CPU traffic. For our benchmarks we can assume different values of CPU bound traffic. The different percentage of CPU bound traffic must be specified in the benchmark.

4. Network Energy Consumption Rate (NECR)

To optimize the run time energy usage for different devices, the additional energy consumption that will result as a factor of additional traffic needs to be known. The NECR defines the power usage increase in MilliWatts per Mbps of data at the physical layer.

The NECR will depend on the line card, the port and the other factors defined earlier.

For the effective use of the NECR the base power of the chassis, a line card and a port needs to be specified when there is no load. The measurements must take into consideration power optimization techniques when there is no traffic on any port of a line card.

5. Network Energy Proportionality Index (NEPI)

In the ideal case the power consumed by a device is proportional to its network load. The average difference between the ideal(I) and the measured (M) power consumption defines the EPI.

The ideal power is measured by assuming the power consumed by a device at 100% traffic load and using that to derive the ideal power usage for different traffic loads.

$$EPI_x = (M_x - I_x) / M_x * 100$$

$$EPI = EPI_1 + EPI_2 + \dots + EPI_n / n$$

The EPI is independent of the actual traffic load. It can thus be used to define the energy efficiency of a networking device. A value of 0 means the power usage is agnostic to traffic and a value of 100 means that the device has perfect energy proportionality.

6. Benchmark Details

All power measurements are done in MilliWatts, except NECR which is done in MilliWatts/ Mbps.

7. Security Considerations

This document raises no new security issues.

8. IANA Considerations

No actions are required from IANA for this informational document.

9. Acknowledgements

This document derives a lot of its text and content from "A Power Benchmarking Framework for Network Devices" paper and the authors of that are duly acknowledged.

The author would like to thank Srini Seetharaman - srini.seetharaman@telekom.com and Priya Mahadevan priya.mahadevan@hp.com for their support with the draft. The author would also like to thank Al Morton - AT&T and Robert Peglar- XioTech for his careful reading and suggestions on the draft.

10. References

10.1. Normative References

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

10.2. Informative References

- [RFC2554] Bradner, S., "Benchmarking Methodology for Network Interconnect Devices", March 1999.

Authors' Addresses

Vishwas Manral
Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304
USA

Email: vishwas.manral@hp.com

Puneet Sharma
Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304
USA

Email: puneet.sharma@hp.com

Yang Ping
H3C.
TBD.
Beijing, CO 12345
China

Email: yangpin@h3c.com

Benchmarking Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 27, 2012

R. Papneja
Huawei Technologies
B. Parise
Cisco Systems
S. Hares
Huawei Technologies
I. Varlashkin
Easynet Global Services
March 26, 2012

Basic BGP Convergence Benchmarking Methodology for Data Plane
Convergence
draft-papneja-bgp-basic-dp-convergence-03.txt

Abstract

BGP is widely deployed and used by several service providers as the default Inter AS routing protocol. It is of utmost importance to ensure that when a BGP peer or a downstream link of a BGP peer fails, the alternate paths are rapidly used and routes via these alternate paths are installed. This document provides the basic BGP Benchmarking Methodology using existing BGP Convergence Terminology, RFC 4098.

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 September 27, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents
(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
1.1. Precise Benchmarking Definition	4
1.2. Purpose of BGP FIB (Data Plane) Convergence	4
1.3. Control Plane Convergence	5
1.4. Benchmarking Testing	5
2. Existing Definitions and Requirements	5
3. Test Topologies	6
3.1. General Reference Topologies	6
4. Test Considerations	8
4.1. Number of Peers	9
4.2. Number of Routes per Peer	9
4.3. Policy Processing/Reconfiguration	9
4.4. Configured Parameters (Timers, etc..)	9
4.5. Interface Types	11
4.6. Measurement Accuracy	11
4.7. Measurement Statistics	11
4.8. Authentication	12
4.9. Convergence Events	12
4.10. High Availability	12
5. Test Cases	12
5.1. Basic Convergence Tests	12
5.1.1. RIB-IN Convergence	13
5.1.2. RIB-OUT Convergence	14
5.1.3. eBGP Convergence	16
5.1.4. iBGP Convergence	16
5.1.5. eBGP Multihop Convergence	16
5.2. BGP Failure/Convergence Events	18
5.2.1. Physical Link Failure on DUT End	18
5.2.2. Physical Link Failure on Remote/Emulator End	19
5.2.3. ECMP Link Failure on DUT End	19
5.3. BGP Adjacency Failure (Non-Physical Link Failure) on Emulator	19
5.4. BGP Hard Reset Test Cases	21
5.4.1. BGP Non-Recovering Hard Reset Event on DUT	21
5.5. BGP Soft Reset	22
5.6. BGP Route Withdrawal Convergence Time	23
5.7. BGP Path Attribute Change Convergence Time	25
5.8. BGP Graceful Restart Convergence Time	26
6. Reporting Format	28
7. IANA Considerations	31
8. Security Considerations	31
9. References	31
9.1. Normative References	31
9.2. Informative References	32
Authors' Addresses	32

1. Introduction

This document defines the methodology for benchmarking data plane FIB convergence performance of BGP in router and switches for simple topologies of 3 or 4 nodes. The methodology proposed in this document applies to both IPv4 and IPv6 and if a particular test is unique to one version, it is marked accordingly. For IPv6 benchmarking the device under test will require the support of Multi-Protocol BGP (MP-BGP) [RFC4760, RFC2545].

The scope of this companion document is limited to basic BGP protocol FIB convergence measurements. BGP extensions outside of carrying IPv6 in (MP-BGP) [RFC4760, RFC2545] are outside the scope of this document. Interaction with IGPs (IGP interworking) is outside the scope of this document.

1.1. Precise Benchmarking Definition

Since benchmarking is science of precision, let us restate the purpose of this document in benchmarking terms. This document defines methodology to test

- data plane convergence on a single BGP device that supports the BGP [RFC4271] functionality
- in test topology of 3 or 4 nodes
- using Basic BGP

Data plane convergence is defined as the completion of all FIB changes so that all forwarded traffic now takes the new proposed route. RFC 4098 defines the terms BGP device, FIB and the forwarded traffic. Data plane convergence is different than control plane convergence within a node.

Basic BGP is defined as RFC 4271 functional with Multi-Protocol BGP (MP-BGP) [RFC4760, RFC2545] for IPv6. The use of other extensions of BGP to support layer-2, layer-3 virtual private networks (VPN) are out of scope of this document.

The terminology used in this document is defined in [RFC4098]. One additional term is defined in this draft: FIB (Data plane) BGP Convergence.

1.2. Purpose of BGP FIB (Data Plane) Convergence

In the current Internet architecture the Inter-Autonomous System (inter-AS) transit is primarily available through BGP. To maintain a

reliable connectivity within intra-domains or across inter-domains, fast recovery from failures remains most critical. To ensure minimal traffic losses, many service providers are requiring BGP implementations to converge the entire Internet routing table within sub-seconds at FIB level.

Furthermore, to compare these numbers amongst various devices, service providers are also looking at ways to standardize the convergence measurement methods. This document offers test methods for simple topologies. These simple tests will provide a quick high-level check, of the BGP data plane convergence across multiple implementations.

1.3. Control Plane Convergence

The convergence of BGP occurs at two levels: RIB and FIB convergence. RFC 4098 defines terms for BGP control plane convergence. Methodologies which test control plane convergence are out of scope for this draft.

1.4. Benchmarking Testing

In order to ensure that the results obtained in tests are repeatable, careful setup of initial conditions and exact steps are required.

This document proposes these initial conditions, test steps, and result checking. To ensure uniformity of the results all optional parameters SHOULD be disabled and all settings SHOULD be changed to default, these may include BGP timers as well.

2. Existing Definitions and Requirements

RFC 1242, "Benchmarking Terminology for Network Interconnect Devices" [RFC1242] and RFC 2285, "Benchmarking Terminology for LAN Switching Devices" [RFC2285] SHOULD be reviewed in conjunction with this document. WLAN-specific terms and definitions are also provided in Clauses 3 and 4 of the IEEE 802.11 standard [802.11]. Commonly used terms may also be found in RFC 1983 [RFC1983].

For the sake of clarity and continuity, this document adopts the general template for benchmarking terminology set out in Section 2 of RFC 1242. Definitions are organized in alphabetical order, and grouped into sections for ease of reference. The following terms are assumed to be taken as defined in RFC 1242 [RFC1242]: Throughput, Latency, Constant Load, Frame Loss Rate, and Overhead Behavior. In addition, the following terms are taken as defined in [RFC2285]: Forwarding Rates, Maximum Forwarding Rate, Loads, Device Under Test

(DUT), and System Under Test (SUT).

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].

3. Test Topologies

This section describes simple test setups for use in BGP benchmarking tests measuring convergence of the FIB (data plane) after the BGP updates has been received.

These simple test nodes have 3 or 4 nodes with the following configuration:

1. Basic Test Setup
2. Three node setup for iBGP or eBGP convergence
3. Setup for eBGP multihop test scenario
4. Four node setup for iBGP or eBGP convergence

Individual tests refer to these topologies.

Figures 1-4 use the following conventions

- o AS-X: Autonomous System X
- o Loopback Int: Loopback interface on the BGP enabled device
- o R2: Helper router

3.1. General Reference Topologies

Emulator acts as 1 or more BGP peers for different testcases.

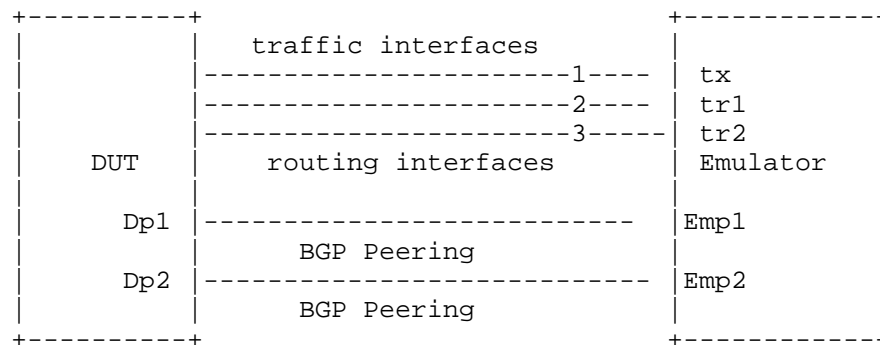


Figure 1 Basic Test Setup

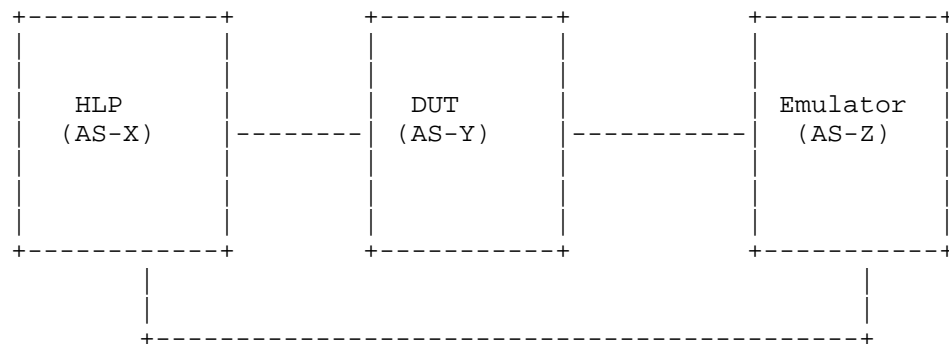


Figure 2 Three Node Setup for eBGP and iBGP Convergence

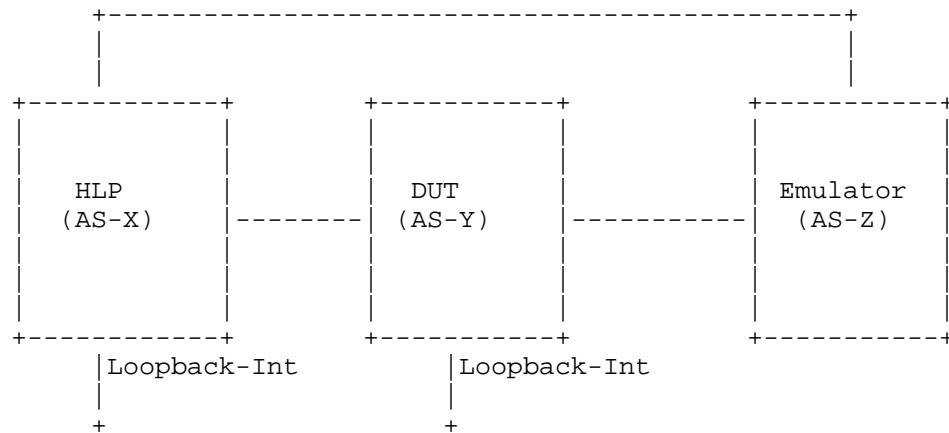


Figure 3 BGP Convergence for eBGP Multihop Scenario

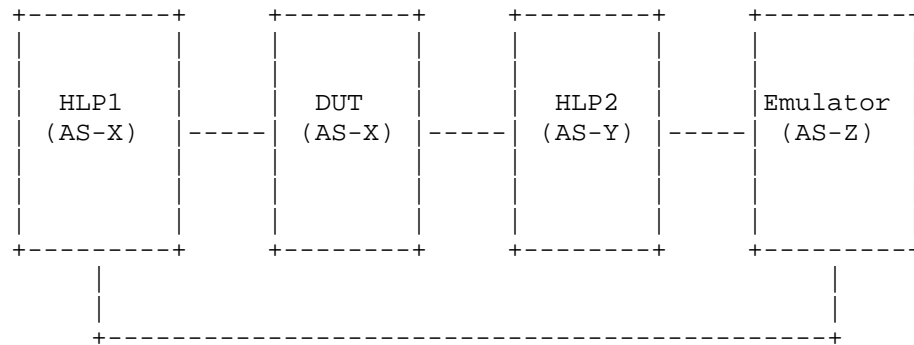


Figure 4 Four Node Setup for EBGP and IBGP Convergence

4. Test Considerations

The test cases for measuring convergence for iBGP and eBGP are different. Both iBGP and eBGP use different mechanisms to advertise, install and learn the routes. Typically, an iBGP route on the DUT is installed and exported when the next-hop is valid. For eBGP the

route is installed on the DUT with the remote interface address as the next-hop with the exception of the multihop case.

4.1. Number of Peers

Number of Peers is defined as the number of BGP neighbors or sessions the DUT has at the beginning of the test. The peers are established before the tests begin. The relationship could be either, iBGP or eBGP peering depending upon the test case requirement.

The DUT establishes one or more BGP sessions with one more emulated routers or helper nodes. Additional peers can be added based on the testing requirements. The number of peers enabled during the testing should be well documented in the report matrix.

4.2. Number of Routes per Peer

Number of Routes per Peer is defined as the number of routes advertized or learnt by the DUT per session or through neighbor relationship with an emulator or helper node. The tester, emulating as neighbor MUST advertise at least one route per peer.

Each test run must identify the route stream in terms of route packing, route mixture, and number of routes. This route stream must be well documented in the reporting stream. RFC 4098 defines these terms.

It is RECOMMENDED that the user may consider advertizing the entire current Internet routing table per peering session using an Internet route mixture with unique or non-unique routes. If multiple peers are used, it is important to precisely document the timing sequence between the peer sending routes (as defined in RFC 4098).

4.3. Policy Processing/Reconfiguration

The DUT MUST run one baseline test where policy is Minimal policy as defined in RFC 4098. Additional runs may be done with policy set-up before the tests begin. Exact policy settings should be documented as part of the test.

4.4. Configured Parameters (Timers, etc..)

There are configured parameters and timers that may impact the measured BGP convergence times.

The benchmark metrics MAY be measured at any fixed values for these configured parameters.

It is RECOMMENDED these configure parameters have the following settings: a) default values specified by the respective RFC b) platform-specific default parameters and c) values as expected in the operational network. All optional BGP settings MUST be kept consistent across iterations of any specific tests

Examples of the configured parameters that may impact measured BGP convergence time include, but are not limited to:

1. Interface failure detection timer
2. BGP Keepalive timer
3. BGP Holdtime
4. BGP update delay timer
5. ConnectRetry timer
6. TCP Segment Size
7. Minimum Route Advertisement Interval (MRAI)
8. MinASOriginationInterval (MAOI)
9. Route Flap Dampening parameters
10. TCP MD5
11. Maximum TCP Window Size
12. MTU

The basic-test settings for the parameters should be:

1. Interface failure detection timer (0 ms)
2. BGP Keepalive timer (1 min)
3. BGP Holdtime (3 min)
4. BGP update delay timer (0 s)

5. ConnectRetry timer (1 s)
6. TCP Segment Size (4096)
7. Minimum Route Advertisement Interval (MRAI) (0 s)
8. MinASOriginationInterval (MAOI)(0 s)
9. Route Flap Dampening parameters (off)
10. TCP MD5 (off)

4.5. Interface Types

The type of media dictate which test cases may be executed, each interface type has unique mechanism for detecting link failures and the speed at which that mechanism operates will influence the measurement results. All interfaces MUST be of the same media and throughput for each test case.

4.6. Measurement Accuracy

Since observed packet loss is used to measure the route convergence time, the time between two successive packets offered to each individual route is the highest possible accuracy of any packet-loss based measurement. When packet jitter is much less than the convergence time, it is a negligible source of error and hence it will be treated as within tolerance.

Other options to measure convergence are the Time-Based Loss Method (TBLM) and Timestamp Based Method(TBM)[MPLSProt]

An exterior measurement on the input media (such Ethernet)is defined by this specification.

4.7. Measurement Statistics

The benchmark measurements may vary for each trial, due to the statistical nature of timer expirations, CPU scheduling, etc. It is recommended to repeat the test multiple times. Evaluation of the test data must be done with an understanding of generally accepted testing practices regarding repeatability, variance and statistical significance of a small number of trials.

For any repeated tests that are averaged to remove variance, all parameters MUST remain the same.

4.8. Authentication

Authentication in BGP is done using the TCP MD5 Signature Option [RFC5925]. The processing of the MD5 hash, particularly in devices with a large number of BGP peers and a large amount of update traffic, can have an impact on the control plane of the device. If authentication is enabled, it SHOULD be documented correctly in the reporting format

4.9. Convergence Events

Convergence events or triggers are defined as abnormal occurrences in the network, which initiate route flapping in the network, and hence forces the re-convergence of a steady state network. In a real network, a series of convergence events may cause convergence latency operators desire to test.

These convergence events must be defined in terms of the sequences defined in RFC 4098. This basic document begins all tests with a router initial set-up. Additional documents will define BGP data plane convergence based on peer initialization.

The convergence events may or may not be tied to the actual failure. A Soft Reset (RFC 4098) does not clear the RIB or FIB tables. A Hard reset clears the BGP peer sessions, the RIB tables, and FIB tables.

4.10. High Availability

Due to the different Non-Stop-Routing (sometimes referred to High-Availability) solutions available from different vendors, it is RECOMMENDED that any redundancy available in the routing processors should be disabled during the convergence measurements.

5. Test Cases

All tests defined under this section assume the following:

- a. BGP peers should be brought to BGP Peer established state
- b. Furthermore the traffic generation and routing should be verified in the topology

5.1. Basic Convergence Tests

These test cases measure characteristics of a BGP implementation in non-failure scenarios like:

1. RIB-IN Convergence
2. RIB-OUT Convergence
3. eBGP Convergence
4. iBGP Convergence

5.1.1. RIB-IN Convergence

Objective:

This test measures the convergence time taken to receive and install a route in RIB using BGP

Reference Test Setup:

This test uses the setup as shown in figure 1

Procedure:

- A. All variables affecting Convergence should be set to a basic test state (as defined in section 4-4).
- B. Establish BGP adjacency between DUT and peer x of Emulator.
- C. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test.
- D. Start the traffic from the Emulator peer-x towards the DUT targeted at a routes specified in route mixture (ex. route A) Initially no traffic SHOULD be observed on the egress interface as the route A is not installed in the forwarding database of the DUT.
- E. Advertise route A from the Peer-x to the DUT and record the time.

This is $Tup(EMx, Rt-A)$ also named 'XMT-Rt-time'.

- F. Record the time when the route-A from Peer-x is received at the DUT.

This $Tup(DUT, Rt-A)$ also named 'RCV-Rt-time'.

- G. Record the time when the traffic targeted towards route A is received by Emulator on appropriate traffic egress interface.

This is $TR(TDr, Rt-A)$. This is also named DUT-XMT-Data-Time.

- H. The difference between the $Tup(DUT, RT-A)$ and traffic received time ($TR(TDr, Rt-A)$) is the FIB Convergence Time for route-A in the route mixture. A full convergence for the route update is the measurement between the 1st route (Route-A) and the last route ($Rt-last$)

Route update convergence is

$TR(TDr, RT-last) - Tup(DUT, Rt-A)$ or

$(DUT-XMT-Data-Time - RCV-Rt-Time)(Rt-A)$

Note: It is recommended that a single test with the same route mixture be repeated several times. A report should provide the Standard Deviation of all tests and the Average.

Running tests with a varying number of routes and route mixtures is important to get a full characterization of a single peer.

5.1.2. RIB-OUT Convergence

Objective:

This test measures the convergence time taken by an implementation to receive, install and advertise a route using BGP

Reference Test Setup:

This test uses the setup as shown in figure 2

Procedure:

- A. The Helper node (HLP) run same version of BGP as DUT.

- B. All devices MUST be synchronized using NTP or some local reference clock.
- C. All configuration variables for HLP, DUT, and Emulator SHOULD be set to the same values. These values MAY be basic-test or a unique set completely described in the test set-up.
- D. Establish BGP adjacency between DUT and Emulator.
- E. Establish BGP adjacency between DUT and Helper Node.
- F. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- G. Start the traffic from the Emulator towards the Helper Node targeted at a specific route say route A. Initially no traffic SHOULD be observed on the egress interface as the route-A is not installed in the forwarding database of the DUT.
- H. Advertise routeA from the Emulator to the DUT and note the time.

This is $Tup(EMx, Route-A)$. (also named EM-XMT-Rt-Time)

- I. Record when Route-A is received by DUT.

This is $Tup(DUTr, Route-A)$. (also named DUT-RCV-Rt-Time)

- J. Record the time when the ROUTE is forwarded by DUT towards the Helper node.

This is $Tup(DUTx, Route-A)$. (also named DUT-XMT-Rt-Time)

- K. Record the time when the traffic targeted towards route-A is received on the Route Egress Interface. This is $TR(EMr, Route-A)$. (also named DUT-XMT-Data Time).

$FIB\ convergence = (DUT-RCV-Rt-Time - DUT-XMT-Data-Time)$

$RIB\ convergence = (DUT-RCV-Rt-Time - DUT-XMT-Rt-Time)$

Convergence for a route stream is characterized by

a) Individual route convergence for FIB, RIB

b) All route convergence of

FIB-convergence =DUT-RCV-Rt-Time(A)-DUT-XMT-Data-Time(last)

RIB-convergence =DUT-RCV-Rt-Time(A)-DUT-XMT-Rt-Time(last)

5.1.3. eBGP Convergence

Objective:

This test measures the convergence time taken by an implementation to receive, install and advertise a route in an eBGP Scenario

Reference Test Setup:

This test uses the setup as shown in figure 2 and the scenarios described in RIB-IN and RIB-OUT are applicable to this test case.

5.1.4. iBGP Convergence

Objective:

This test measures the convergence time taken by an implementation to receive, install and advertise a route in an iBGP Scenario

Reference Test Setup:

This test uses the setup as shown in figure 2 and the scenarios described in RIB-IN and RIB-OUT are applicable to this test case.

5.1.5. eBGP Multihop Convergence

Objective:

This test measures the convergence time taken by an implementation to receive, install and advertise a route in an eBGP Multihop Scenario

Reference Test Setup:

This test uses the setup as shown in figure 3. DUT is used along with a helper node.

Procedure:

- A. The Helper Node (HLP) runs the same BGP version as DUT
- B. All devices to be synchronized using NTP
- C. All variables affecting Convergence like authentication, policies, timers should be set to basic-settings
- D. All 3 devices, DUT, Emulator and Helper Node are configured with different Autonomous Systems
- E. Loopback Interfaces are configured on DUT and Helper Node and connectivity is established between them using any config options available on the DUT
- F. Establish BGP adjacency between DUT and Emulator
- G. Establish BGP adjacency between DUT and Helper Node
- H. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- I. Start the traffic from the Emulator towards the DUT targeted at a specific route say routeA
- J. Initially no traffic SHOULD be observed on the egress interface as the routeA is not installed in the forwarding database of the DUT
- K. Advertise routeA from the Emulator to the DUT and note the time (Tup(EMx,RouteA) also named (Route-Tx-time)
- L. Record the time when the route is received by the DUT. This is Tup(EMr,DUT) named (Route-Rcv-time)
- M. Record the time when the traffic targeted towards routeA is received from Egress Interface of DUT on emulator. This is Tup(EMd,DUT) named (Data-Rcv-time)
- N. Record the time when the routeA is forwarded by DUT towards the Helper node. This is Tup(EMf,DUT) also named (Route-Fwd-time)

FIB Convergence = (Data-Rcv-time - Route-Rcv-time)

RIB Convergence = (Route-Fwd-time - Route-Rcv-time)

Note: It is recommended that the test be repeated with varying number

of routes and route mixtures. With each set route mixture, the test should be repeated multiple times. The results should record average, mean, Standard Deviation

5.2. BGP Failure/Convergence Events

5.2.1. Physical Link Failure on DUT End

Objective:

This test measures the route convergence time due to local link failure event at DUT's Local Interface

Reference Test Setup:

This test uses the setup as shown in figure 1. Shutdown event is defined as an administrative shutdown event on the DUT

Procedure:

- A. All variables affecting Convergence like authentication, policies, timers should be set to basic-test policy
- B. Establish 2 BGP adjacencies from DUT to Emulator, one over the peer interface and the other using a second peer interface
- C. Advertise the same route, route A over both the adjacencies and (Tx1)Interface to be the preferred next hop
- D. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- E. Start the traffic from the Emulator towards the DUT targeted at a specific route say route A. Initially traffic would be observed on the best egress route (Emp1) instead of Trr2
- F. Trigger the shutdown event of Best Egress Interface on DUT (Drr1)
- G. Measure the Convergence Time for the event to be detected and traffic to be forwarded to Next-Best Egress Interface (rr2)

Time = Data-detect(rr2) - Shutdown time

H. Stop the offered load and wait for the queues to drain and Restart

I. Bring up the link on DUT Best Egress Interface

J. Measure the convergence time taken for the traffic to be rerouted from (rr2) to Best Interface (rr1)

Time = Data-detect(rr1) - Bring Up time

K. It is recommended that the test be repeated with varying number of routes and route mixtures or with number of routes & route mixtures closer to what is deployed in operational networks

5.2.2. Physical Link Failure on Remote/Emulator End

Objective:

This test measures the route convergence time due to local link failure event at Tester's Local Interface

Reference Test Setup:

This test uses the setup as shown in figure 1. Shutdown event is defined as shutdown of the local interface of Tester via logical shutdown event. The procedure used in 5.2.1 is used for the termination

5.2.3. ECMP Link Failure on DUT End

Objective:

This test measures the route convergence time due to local link failure event at ECMP Member. The FIB configuration and BGP is set to allow two ECMP routes to be installed. However, policy directs the routes to be sent only over one of the paths

Reference Test Setup:

This test uses the setup as shown in figure 1 and the procedure uses 5.2.1

5.3. BGP Adjacency Failure (Non-Physical Link Failure) on Emulator

Objective:

This test measures the route convergence time due to BGP Adjacency Failure on Emulator

Reference Test Setup:

This test uses the setup as shown in figure 1

Procedure:

- A. All variables affecting Convergence like authentication, policies, timers should be basic-policy set
- B. Establish 2 BGP adjacencies from DUT to Emulator, one over the Best Egress Interface and the other using the Next-Best Egress Interface
- C. Advertise the same route, routeA over both the adjacencies and make Best Egress Interface to be the preferred next hop
- D. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- E. Start the traffic from the Emulator towards the DUT targeted at a specific route say routeA. Initially traffic would be observed on the Best Egress interface
- F. Remove BGP adjacency via a software adjacency down on the Emulator on the Best Egress Interface. This time is called BGPadj-down-time also termed BGPpeer-down
- G. Measure the Convergence Time for the event to be detected and traffic to be forwarded to Next-Best Egress Interface. This time is Tr-rr2 also called TR2-traffic-on

$$\text{Convergence} = \text{TR2-traffic-on} - \text{BGPpeer-down}$$

- H. Stop the offered load and wait for the queues to drain and Restart
- I. Bring up BGP adjacency on the Emulator over the Best Egress Interface. This time is BGP-adj-up also called BGPpeer-up
- J. Measure the convergence time taken for the traffic to be rerouted to Best Interface. This time is BGP-adj-up also called BGPpeer-up

5.4. BGP Hard Reset Test Cases

5.4.1. BGP Non-Recovering Hard Reset Event on DUT

Objective:

This test measures the route convergence time due to Hard Reset on the DUT

Reference Test Setup:

This test uses the setup as shown in figure 1

Procedure:

- A. The requirement for this test case is that the Hard Reset Event should be non-recovering and should affect only the adjacency between DUT and Emulator on the Best Egress Interface
- B. All variables affecting SHOULD be set to basic-test values
- C. Establish 2 BGP adjacencies from DUT to Emulator, one over the Best Egress Interface and the other using the Next-Best Egress Interface
- D. Advertise the same route, routeA over both the adjacencies and make Best Egress Interface to be the preferred next hop
- E. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- F. Start the traffic from the Emulator towards the DUT targeted at a specific route say routeA. Initially traffic would be observed on the Best Egress interface
- G. Trigger the Hard Reset event of Best Egress Interface on DUT
- H. Measure the Convergence Time for the event to be detected and traffic to be forwarded to Next-Best Egress Interface

Time of convergence = time-traffic flow - time-reset

- I. Stop the offered load and wait for the queues to drain and Restart
- J. It is recommended that the test be repeated with varying number of routes and route mixtures or with number of routes & route mixtures closer to what is deployed in operational networks
- K. When varying number of routes are used, convergence Time is measured using the Loss Derived method [IGPData]
- L. Convergence Time in this scenario is influenced by Failure detection time on Tester, BGP Keep Alive Time and routing, forwarding table update time

5.5. BGP Soft Reset

Objective:

This test measures the route convergence time taken by an implementation to service a BGP Route Refresh message and advertise a route

Reference Test Setup:

This test uses the setup as shown in figure 2

Procedure:

- A. The BGP implementation on DUT & Helper Node needs to support BGP Route Refresh Capability [RFC2918]
- B. All devices to be synchronized using NTP
- C. All variables affecting Convergence like authentication, policies, timers should be set to basic-test defaults
- D. DUT and Helper Node are configured in the same Autonomous System whereas Emulator is configured under a different Autonomous System
- E. Establish BGP adjacency between DUT and Emulator
- F. Establish BGP adjacency between DUT and Helper Node

- G. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- H. Configure a policy under BGP on Helper Node to deny routes received from DUT
- I. Advertise routeA from the Emulator to the DUT
- J. The DUT will try to advertise the route to Helper Node will be denied
- K. Wait for 3 KeepAlives
- L. Start the traffic from the Emulator towards the Helper Node targeted at a specific route say routeA. Initially no traffic would be observed on the Egress interface, as routeA is not present
- M. Remove the policy on Helper Node and issue a Route Refresh request towards DUT. Note the timestamp of this event. This is the RefreshTime
- N. Record the time when the traffic targeted towards routeA is received on the Egress Interface. This is RecTime
- O. The following equation represents the Route Refresh Convergence Time per route

$$\text{Route Refresh Convergence Time} = (\text{RecTime} - \text{RefreshTime})$$

5.6. BGP Route Withdrawal Convergence Time

Objective:

This test measures the route convergence time taken by an implementation to service a BGP Withdraw message and advertise the withdraw

Reference Test Setup:

This test uses the setup as shown in figure 2

Procedure:

- A. This test consists of 2 steps to determine the Total Withdraw Processing Time
- B. Step 1:
- (1) All devices to be synchronized using NTP
 - (2) All variables should be set to basic-test parameters
 - (3) DUT and Helper Node are configured in the same Autonomous System whereas Emulator is configured under a different Autonomous System
 - (4) Establish BGP adjacency between DUT and Emulator
 - (5) To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
 - (6) Start the traffic from the Emulator towards the DUT targeted at a specific route say routeA. Initially no traffic would be observed on the Egress interface as the routeA is not present on DUT
 - (7) Advertise routeA from the Emulator to the DUT
 - (8) The traffic targeted towards routeA is received on the Egress Interface
 - (9) Now the Tester sends request to withdraw routeA to DUT, TRx(Awith) also called WdrawTime1
 - (10) Record the time when no traffic is observed on the Egress Interface. This is the RouteRemoveTime1(A)

WdrawConvTime1 = RouteRemoveTime1(A)
 - (11) The difference between the RouteRemoveTime1 and WdrawTime1 is the WdrawConvTime1
- C. Step 2:
- (1) Continuing from Step 1, re-advertise routeA back to DUT from Tester

- (2) The DUT will try to advertise the routeA to Helper Node (assumption there exists a session between DUT and helper node)
- (3) Start the traffic from the Emulator towards the Helper Node targeted at a specific route say routeA. Traffic would be observed on the Egress interface after routeA is received by the Helper Node

WATime=time traffic first flows

- (4) Now the Tester sends a request to withdraw routeA to DUT. This is the WdrawTime2

WAWtime-TRx(RouteA) = WdrawTime2

- (5) DUT processes the withdraw and sends it to Helper Node
- (6) Record the time when no traffic is observed on the Egress Interface of Helper Node. This is

TR-WAW(DUT,RouteA) = RouteRemoveTime2

- (7) Total withdraw processing time is

TotalWdrawTime = ((RouteRemoveTime2 - WdrawTime2) - WdrawConvTime1)

5.7. BGP Path Attribute Change Convergence Time

Objective:

This test measures the convergence time taken by an implementation to service a BGP Path Attribute Change

Reference Test Setup:

This test uses the setup as shown in figure 1

Procedure:

- A. This test only applies to Well-Known Mandatory Attributes like Origin, AS Path, Next Hop
- B. In each iteration of test only one of these mandatory attributes need to be varied whereas the others remain the

same

- C. All devices to be synchronized using NTP
- D. All variables should be set to basic-test parameters
- E. Advertise the route, routeA over the Best Egress Interface only, making it the preferred named Tbest
- F. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- G. Start the traffic from the Emulator towards the DUT targeted at the specific route say routeA. Initially traffic would be observed on the Best Egress interface
- H. Now advertise the same route routeA on the Next-Best Egress Interface but by varying one of the well-known mandatory attributes to have a preferred value over that interface. We call this Tbetter. The other values need to be same as what was advertised on the Best-Egress adjacency

$TRx(\text{Path-Change}) = \text{Path Change Event Time}$

- I. Measure the Convergence Time for the event to be detected and traffic to be forwarded to Next-Best Egress Interface

$DUT(\text{Path-Change}, \text{RouteA}) = \text{Path-switch time}$

$\text{Convergence} = \text{Path-switch time} - \text{Path Change Event Time}$

- J. Stop the offered load and wait for the queues to drain and Restart
- K. Repeat the test for various attributes

5.8. BGP Graceful Restart Convergence Time

Objective:

This test measures the route convergence time taken by an implementation during a Graceful Restart Event

Reference Test Setup:

This test uses the setup as shown in figure 4

Procedure:

- A. It measures the time taken by an implementation to service a BGP Graceful Restart Event and advertise a route
- B. The Helper Nodes are the same model as DUT and run the same BGP implementation as DUT
- C. The BGP implementation on DUT & Helper Node needs to support BGP Graceful Restart Mechanism [RFC4724]
- D. All devices to be synchronized using NTP
- E. All variables are set to basic-test values
- F. DUT and Helper Node-1(HLP1) are configured in the same Autonomous System whereas Emulator and Helper Node-2(HLP2) are configured under different Autonomous Systems
- G. Establish BGP adjacency between DUT and Helper Nodes
- H. Establish BGP adjacency between Helper Node-2 and Emulator
- I. To ensure adjacency establishment, wait for 3 KeepAlives from the DUT or a configurable delay before proceeding with the rest of the test
- J. Configure a policy under BGP on Helper Node-1 to deny routes received from DUT
- K. Advertise routeA from the Emulator to Helper Node-2
- L. Helper Node-2 advertises the route to DUT and DUT will try to advertise the route to Helper Node-1 which will be denied
- M. Wait for 3 KeepAlives
- N. Start the traffic from the Emulator towards the Helper Node-1 targeted at the specific route say routeA. Initially no traffic would be observed on the Egress interface as the routeA is not present
- O. Perform a Graceful Restart Trigger Event on DUT and note the time. This is the GREventTime

- P. Remove the policy on Helper Node-1
- Q. Record the time when the traffic targeted towards routeA is received on the Egress Interface

TRr(DUT, routeA). This is also called RecTime
- R. The following equation represents the Graceful Restart Convergence Time

$$\text{Graceful Restart Convergence Time} = ((\text{RecTime} - \text{GREventTime}) - \text{RIB-IN})$$
- S. It is assumed in this test case that after a Switchover is triggered on the DUT, it will not have any cycles to process BGP Refresh messages. The reason for this assumption is that there is a narrow window of time where after switchover when we remove the policy from Helper Node -1, implementations might generate Route-Refresh automatically and this request might be serviced before the DUT actually switches over and reestablishes BGP adjacencies with the peers

6. Reporting Format

For each test case, it is recommended that the reporting tables below are completed and all time values SHOULD be reported with resolution as specified in [RFC4098]

Parameter	Units
Test case	Test case number
Test topology	1,2,3 or 4
Parallel links	Number of parallel links
Interface type	GigE, POS, ATM, other
Convergence Event	Hard reset, Soft reset, link failure, or other defined
eBGP sessions	Number of eBGP sessions
iBGP sessions	Number of iBGP sessions
eBGP neighbor	Number of eBGP neighbors
iBGP neighbor	Number of iBGP neighbors
Routes per peer	Number of routes
Total unique routes	Number of routes
Total non-unique routes	Number of routes
IGP configured	ISIS, OSPF, static, or other
Route Mixture	Description of Route mixture
Route Packing	Number of routes in an update
Policy configured	Yes, No
Packet size offered to the DUT	Bytes
Offered load	Packets per second
Packet sampling interval on tester	Seconds
Forwarding delay threshold	Seconds
Timer Values configured on DUT	
Interface failure indication delay	Seconds
Hold time	Seconds
MinRouteAdvertisementInterval (MRAI)	Seconds
MinASOriginationInterval (MAOI)	Seconds
Keepalive Time	Seconds
ConnectRetry	Seconds
TCP Parameters for DUT and tester	
MSS	Bytes
Slow start threshold	Bytes
Maximum window size	Bytes

Test Details:

- a. If the Offered Load matches a subset of routes, describe how this subset is selected
- b. Describe how the Convergence Event is applied; does it cause instantaneous traffic loss or not

c. If there is any policy configured, describe the configured policy

Complete the table below for the initial Convergence Event and the reversion Convergence Event

Parameter	Unit
Convergence Event	Initial or reversion
Traffic Forwarding Metrics	
Total number of packets offered to DUT	Number of packets
Total number of packets forwarded by DUT	Number of packets
Connectivity Packet Loss	Number of packets
Convergence Packet Loss	Number of packets
Out-of-order packets	Number of packets
Duplicate packets	Number of packets
Convergence Benchmarks	
Rate-derived Method [IGP-Data]:	
First route convergence time	Seconds
Full convergence time	Seconds
Loss-derived Method [IGP-Data]:	
Loss-derived convergence time	Seconds
Route-Specific Loss-Derived Method:	
Minimum R-S convergence time	Seconds
Maximum R-S convergence time	Seconds
Median R-S convergence time	Seconds
Average R-S convergence time	Seconds
Loss of Connectivity Benchmarks	
Loss-derived Method:	
Loss-derived loss of connectivity period	Seconds
Route-Specific loss-derived Method:	
Minimum LoC period [n]	Array of seconds
Minimum Route LoC period	Seconds
Maximum Route LoC period	Seconds
Median Route LoC period	Seconds

Average Route LoC period Seconds

7. IANA Considerations

This draft does not require any new allocations by IANA.

8. 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.

9. References

9.1. Normative References

- [I-D.ietf-bmwg-igp-dataplane-conv-term]
Poretsky, S., Imhoff, B., and K. Michielsen, "Terminology for Benchmarking Link-State IGP Data Plane Route Convergence", draft-ietf-bmwg-igp-dataplane-conv-term-23 (work in progress), February 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918, September 2000.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

9.2. Informative References

- [RFC1242] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [RFC1983] Malkin, G., "Internet Users' Glossary", RFC 1983, August 1996.
- [RFC2285] Mandeville, R., "Benchmarking Terminology for LAN Switching Devices", RFC 2285, February 1998.
- [RFC2545] Marques, P. and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, March 1999.
- [RFC4098] Berkowitz, H., Davies, E., Hares, S., Krishnaswamy, P., and M. Lepp, "Terminology for Benchmarking BGP Device Convergence in the Control Plane", RFC 4098, June 2005.
- [RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, January 2007.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

Authors' Addresses

Rajiv Papneja
Huawei Technologies

Email: rajiv.papneja@huawei.com

Bhavani Parise
Cisco Systems

Email: bhavani@cisco.com

Susan Hares
Huawei Technologies (USA)

Email: shares@huawei.com

Ilya Varlashkin
Easynet Global Services

Email: ilya.varlashkin@easynet.com

Dean Lee
Ixia

Email: dlee@ixiacom.com

Eric Brendel
Independent Consultant

Email: brendel@pektel.com

Mohan Nanduri
Microsoft

Email: mnanduri@microsoft.com

Jay Karthik
Cisco Systems

Email: jkarthik@cisco.com

