

Benchmarking Methodology WG
Internet Draft
Intended status: Informational
Expires: November 21, 2013

Sarah Banks
Aerohive Networks
Fernando Calabria
Cisco
Gery Czirjak
Ramdas Machat
Juniper
June 3, 2013

ISSU Benchmarking Methodology
draft-banks-bmwg-issu-meth-01

Abstract

Modern forwarding devices attempt to minimize any control and data plane disruptions while performing planned software changes, by implementing a technique commonly known as an In Service Software Upgrade (ISSU).

This document specifies a set of common methodologies and procedures designed to characterize the overall behavior of a Device Under Test (DUT) subject to an ISSU event.

Status of this Memo

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 March 2012.

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. Conventions used in this document.....	4
3. Generic ISSU process, phased approach.....	5
3.1. Software Download.....	6
3.2. Software Staging.....	6
3.3. Upgrade Run.....	7
3.4. Upgrade Acceptance.....	7
4. Test Methodology.....	8
5. ISSU Test Methodology.....	10
5.2 Software Staging.....	10

5.3	Upgrade Run.....	11
5.4	Post ISSU verifications.....	12
6	ISSU Abort and Rollback.....	13
7	Final Report - Data Presentation - Analysis.....	14
8	Security Considerations.....	16
9	IANA Considerations.....	16
10	Conclusions.....	16
11	References.....	16
11.1	Normative References.....	16
11.2	Informative References.....	16
12	Acknowledgments.....	16

1. Introduction

ISSU is a technique implemented by forwarding devices to upgrade or downgrade from one software version to another as applicable. The end goal of the entire process is to minimize downtime and/or degradation of service. The ISSU operation may apply in terms of an atomic version change of the entire system software or it may be applied in a more modular sense such as for a patch or maintenance upgrade. The procedure described herein may be used to verify either approach, as may be supported by the vendor hardware and software.

In support of this document, a set of expectations for an ISSU operation can be summarized as follows:

- the software is successfully migrated, from one version to a successive version;
- there are no control plane interruptions throughout the process. That is, the upgrade/downgrade could be accomplished while the device remains "in service". It is noted however, that most service providers will still undertake such actions in a maintenance window (even in redundant environments) to minimize any risk;
- interruptions to the forwarding plane are expected to be minimal to none;
- the total time to accomplish the upgrade is minimized, again to reduce potential network outage exposure (e.g. an external

failure event might impact the network as it operates with reduced redundancy)

This document provides a set of procedures to characterize a given product's ISSU behavior, from the perspective of meeting the above expectations.

Different hardware configurations may be expected to be benchmarked, but a typical configuration for a forwarding device that supports ISSU consists of at least one pair of Routing Processors (RP's) that may operate in a redundant fashion, and single or multiple Forwarding Engines (Line Cards) that may or may not be redundant, as well as fabric cards or other components as applicable. However, this does not preclude the possibility that a device in question can perform ISSU functions through the operation of independent process components, which may be upgraded without impact to the overall operation of the device. As an example, perhaps the software module involved in SNMP functions can be upgraded without impacting other operations.

The concept of a multi-chassis deployment may also be characterized by the current set of proposed methodologies, but the implementation specific details (i.e. process placement and others) are beyond the scope of the current document.

Since most modern forwarding devices, where ISSU would be applicable, do consist of redundant RP's and hardware-separated control plane and data plane functionality, this document will focus on methodologies which would be directly applicable to those platforms. It is anticipated that the concepts and approaches described herein may be readily extended to accommodate other device architectures as well.

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

3. Generic ISSU process, phased approach.

ISSU may be viewed as the behavior of a device when exposed to a planned change in its software functionality. This may mean changes to the core operating system, separate processes or daemons or even of firmware logic in programmable hardware devices (e.g. CPLD/FPGA). The goal of an ISSU implementation is to permit such actions with minimal or no disruption to the primary operation of the device in question.

ISSU may be user initiated through direct interaction with the device or activated through some automated process on a management system or even on the device itself. For the purposes of this document, we will focus on the model where the ISSU action is initiated by direct user intervention.

The ISSU process can be viewed as a series of different phases or activities, as defined below. For each of these phases, the test operator MUST record the outcome as well as any relevant observations (defined further in the present document). Note that, a given vendor implementation may or may not permit the abortion of the in-progress ISSU at particular stages. There may also be certain restrictions as to ISSU availability given certain functional configurations (for example, ISSU in the presence of BiDirectional Failure Detection (BFD) [RFC 5880] may not be supported. It is incumbent upon the test operator to ensure that the DUT is appropriately configured to provide the appropriate test environment as needed. As with any properly orchestrated test effort, the test plan document should reflect these and other relevant details and SHOULD be written with close attention to the expected production-operating environment. The combined analysis of the results of each phase will characterize the overall ISSU process with the main goal of being able to identify and quantify any disruption in service

(from the data and control plane perspective) allowing operators to plan their maintenance activities with greater precision.

The generic ISSU process can be viewed as a series of the following phases:

3.1. Software Download

In this first phase, the requested software package may be downloaded to the router and is typically stored onto a device. This process may be performed automatically by the router as part of the upgrade process, or it may be initiated separately. Such separation allows an administrator to download the new code inside or outside of a maintenance window; it is anticipated that downloading new code and saving it to disk on the router will not impact operations. In the case where the software can be downloaded outside of the actual upgrade process, the administrator SHOULD do so; downloading software can skew timing results based on factors that are often not comparative in nature. Internal compatibility verification may be performed by the software running on the DUT, to verify the checksum of the files downloaded as well as any other pertinent checks. Depending upon vendor implementation, these mechanisms may extend to include verification that the downloaded module(s) meet a set of identified pre-requisites such as hardware or firmware compatibility or minimum software requirements. Where such mechanisms are made available by the product, they should be verified, by the tester, with the perspective of avoiding operational issues in production. Verification should include both positive verification (ensuring that an ISSU action should be permitted) as well as negative tests (creation of scenarios where the verification mechanisms would report exceptions).

3.2. Software Staging

In this second phase, the requested software package is loaded into the pertinent components of a given forwarding device (typically the RP in standby state). Internal compatibility verification may be performed by the software running on the DUT, as part of the upgrade process itself, to verify the checksum of the files downloaded as well as any other pertinent checks. Depending upon vendor implementation, these mechanisms may extend to include verification that the downloaded module(s) meet a set of identified pre-requisites such as hardware or firmware compatibility or minimum

software requirements. Where such mechanisms are made available by the product, they should be verified, by the tester, with the perspective of avoiding operational issues in production. In this case, the execution of these checks is within scope of the upgrade time, and SHOULD be included in the testing results. Once the new software is downloaded to the pertinent components of the DUT, the upgrade begins and the DUT begins to prepare itself for upgrade. Depending on the vendor implementation, it is expected that redundant hardware pieces within the DUT are upgraded, including the backup or secondary RP.

3.3. Upgrade Run

In this phase, the secondary RP takes over, forcing the RP which was previously designated as primary, to adopt the standby role. At this point, the new primary RP drives the required updates to other specific components and forces warm-updates or re-initializations with the new software, as applicable. In addition, the now-standby RP will be updated with the desired software.

This is the critical phase of the ISSU, where the control plane should not be impacted and any interruptions to the forwarding plane should be minimal to none.

For some implementations, the above two steps may be concatenated into one monolithic operation. In such case, the calculation of the respective ISSU time intervals may need to be adapted accordingly. If any control or data plane interruptions occur, it is expected to be observed and recorded within this stage.

3.4. Upgrade Acceptance

In this phase, the new version of software MUST be running in all the physical nodes of the logical forwarding device. (RP's and LC's as applicable). At this point, configuration control is returned to the operator and normal device operation i.e. outside of ISSU-oriented operation, is resumed.

4. Test Methodology

As stated by <http://tools.ietf.org/wg/bmwg/draft-ietf-bmwg-2544-as/> (when it becomes an RFC) The Test Topology Setup must be part of an ITE (Isolated Test Environment)

The reporting of results MUST take into account the repeatability considerations from Section 4 of [RFC2544]. It is RECOMMENDED to perform multiple trials and report average results. The results are reported in a simple statement including the measured frame loss and ISSU impact times.

4.1 Test Topology

The hardware configuration of the DUT (Device Under test) MUST be identical to the one expected to be or currently deployed in production in order for the benchmark to have relevance. This would include the number of RP's, hardware version, memory and initial software release, any common chassis components, such as fabric hardware in the case of a fabric-switching platform and the specific LC's (version, memory, interfaces type, rate etc.)

For the Control and Data plane, differing configuration approaches MAY be utilized. The recommended approach relies on "mimicking" the existing production data and control plane information, in order to emulate all the necessary Layer1 through Layer3 and, if appropriate, upper layer characteristics of the network, as well as end to end traffic/communication pairs. In other words, design a representative load model of the production environment and deploy a collapsed topology utilizing test tools and/or external devices, where the DUT will be tested. Note that, the negative impact of ISSU operations is likely to impact scaled, dynamic topologies to a greater extent than simpler, static environments. As such, this methodology is advised for most test scenarios.

The second, more simplistic approach is to deploy an ITE "Isolated Testing Environment" as described in some of the existing standards for benchmarking methodologies (e.g. RFC2544/RFC6815) in which end-points are "directly" connected to the DUT. In this manner control plane information is kept to a minimum (only connected interfaces) and only a basic data plane of sources and destinations is applied. If this methodology is selected, care must be taken to understand that the systemic behavior of the ITE may not be identical to that

experienced by a device in a production network role. That is, control plane validation may be minimal to none if this methodology is employed. It may be possible to perform some degree of data plane validation with this approach.

4.2 Load Model

In consideration of the defined test topology, a load model must be developed to exercise the DUT while the ISSU event is introduced. This applied load should be defined in such a manner as to provide a granular, repeatable verification of the ISSU impact on transit traffic. Sufficient traffic load (rate) should be applied to permit timing extrapolations at a minimum granularity of 100 milliseconds e.g. 100Mbps for a 10Gbps interface. The use of steady traffic streams rather than bursty loads is preferred to simplify analysis. The traffic should be patterned to provide a broad range of source and destination pairs, which resolve to a variety of FIB (forwarding information base) prefix lengths. If the production network environment includes multicast traffic or VPN's (L2, L3 or IPsec) it is critical to include these in the model.

For mixed protocol environments (e.g. IPv4 and IPv6), frames SHOULD be distributed between the different protocols. The distribution SHOULD approximate the network conditions of deployment. In all cases, the details of the mixed protocol distribution MUST be included in the reporting.

It is recommended that an NMS system be deployed, preferably similar to that utilized in production. This will allow for monitoring of the DUT while it is being tested both in terms of supporting the system resource impact analysis as well as from the perspective of detecting interference with non-transit (management) traffic as a result of the ISSU operation. Additionally, a DUT management session other than snmp-based, typical of usage in production, should be established to the DUT and monitored for any disruption.

It is suggested that the actual test exercise be managed utilizing direct console access to the DUT, if at all possible to avoid the possibility that a network interruption impairs execution of the test exercise.

All in all, the load model should attempt to simulate the production network environment to the greatest extent possible in order to maximize the applicability of the results generated.

5. ISSU Test Methodology

As previously described, for the purposes of this test document, the ISSU process is divided into three main phases. The following methodology assumes that a suitable test topology has been constructed per section 4. A description of the methodology to be applied for each of the above phases follows:

5.1 Pre-ISSU recommended verifications

Verify that enough hardware and software resources are available to complete the Load operation (enough disk space)

Verify that the redundancy states between RPs and other nodes are as expected (e.g. redundancy on, RP's synchronized)

Verify that the device, if running NSR capable routing protocols, is in a "ready" state; that is, that the sync between RPs is complete and the system is ready for failover, if necessary.

Gather a configuration snapshot of the device and all of its applicable components

Verify that the node is operating in a "steady" state (that is, no critical or maintenance function is being currently performed)

Note any other operational characteristics that the tester may deem applicable to the specific implementation deployed.

5.2 Software Staging

Establish all relevant protocol adjacencies and stabilize routing within the test topology. In particular, ensure that the scaled levels of the dynamic protocols are dimensioned as specified by the test topology plan.

Clear relevant logs and interface counters to simplify analysis. If possible, set logging timestamps to a highly granular mode. If the topology includes management systems, ensure that the appropriate polling levels have been applied, sessions established and that the responses are per expectation.

Apply the traffic loads as specified in the load model previously developed for this exercise.

Document an operational baseline for the test bed with relevant data supporting the above steps (include all relevant load characteristics of interest in the topology e.g. routing load, traffic volumes, memory and CPU utilization)

Note the start time (T0) and begin the code change process utilizing the appropriate mechanisms as expected to be used in production (e.g. active download with TFTP/FTP/SCP/etc. or direct install from local or external storage facility). In order to ensure that ISSU process timings are not skewed by the lack of a network wide synchronization source, the use of a network NTP source is encouraged.

Take note of any logging information and command line interface (CLI) prompts as needed (this detail will be vendor-specific). Respond to any DUT prompts in a timely manner.

Monitor the DUT for the reload of secondary RP to the new software level. Once the secondary has stabilized on the new code, note the completion time. The duration of these steps will be logged as "T1".

Review system logs for any anomalies, check that relevant dynamic protocols have remained stable and note traffic loss if any. Verify that deployed management systems have not identified any unexpected behavior.

5.3 Upgrade Run

The following assumes that the software load step and upgrade step are discretely controllable. If not, maintain the afore-mentioned timer and monitor for completion of the ISSU as described below.

Note the start time and initiate the actual upgrade procedure. Monitor the operation of the secondary route processor while it initializes with the new software and assumes mastership of the DUT.

At this point, pay particular attention to any indications of control plane disruption, traffic impact or other anomalous behavior. Once the DUT has converged upon the new code and returned to normal operation note the completion time and log the duration of this step as T2.

Review the syslog data in the DUT and neighboring devices for any behavior, which would be disruptive in a production environment (linecard reloads, control plane flaps etc.). Examine the traffic generators for any indication of traffic loss over this interval. If the Test Set reported any traffic loss, note the number of frames lost as "TP_frames". If the test set also provides outage duration, note this as TP_time (alternatively this may be calculated as TP/offered pps (packets per second) load).

Verify the DUT status observations as per any NMS systems managing the DUT and its neighboring devices. Document the observed CPU and memory statistics both during the ISSU upgrade event and after and ensure that memory and CPU have returned to an expected (previously baselined) level.

5.4 Post ISSU verifications

The following describes a set of post-ISSU verification tasks, that are not directly part of the ISSU process, but are recommended for execution in order to validate a successful upgrade;

- . Configuration delta analysis

- o Examine the post-ISSU configurations to determine if any changes have occurred either through process error or due to differences in the implementation of the upgraded code
- . Exhaustive control plane analysis
 - o Review the details of the RIB and FIB to assess whether any unexpected changes have been introduced in the forwarding paths
- . Verify that both RPs are up and that the redundancy mechanism for the control plane is enabled and fully synchronized.
- . Verify that no control plane (protocol) events or flaps were detected
- . Verify that no L1 and or L2 interface flaps were observed
- . Document the hitless operation or presence of an outage based upon the counter values provided by the Test Set

6 ISSU Abort and Rollback

Where a vendor provides such support, the ISSU process could be aborted for any reason by the operator. However, the end results and behavior may depend on the specific phase where the process was aborted. While this is implementation dependent, as a general recommendation, if the process is aborted during the "Software Download" or "Software Staging" phases, no impact to service or device functionality should be observed. In contrast, if the process is aborted during the "Upgrade Run" or "Upgrade Accept" phases, the system may reload and revert back to the previous software release and as such, this operation may be service affecting.

Where vendor support is available, the abort/rollback functionality should be verified and the impact, if any, quantified generally following the procedures provided above.

7 Final Report - Data Presentation - Analysis

All ISSU impact results are summarized in a simple statement describing the "ISSU Disruption Impact" including the measured frame loss and impact time, where impact time is defined as the time frame determined per the TP reported outage. These are considered to be the primary data points of interest.

However, the entire ISSU operational impact should also be considered in support of planning for maintenance and as such, additional reporting points are included.

Software download/secondary update	T1
Upgrade/Run	T2
ISSU Traffic Disruption (Frame Loss)	TP_frames
ISSU Traffic Impact Time (milliseconds)	TP Time
ISSU Housekeeping Interval	T3
(Time for both RP's up on new code and fully synced - Redundancy restored)	
Total ISSU Maintenance Window	T4 (sum of T1+T2+T3)

The results reporting MUST provide the following information:

- . DUT hardware and software detail
- . Test Topology definition and diagram (especially as related to the ISSU operation)
- . Load Model description including protocol mixes
- . Time Results as per above
- . Anomalies Observed during ISSU
- . Anomalies Observed in post-ISSU analysis

It is RECOMMENDED that the following parameters be reported in these units:

Parameter	Units or Examples
Traffic Load	Frames per second and bits per Second
Disruption (average)	Frames
Impact Time (average)	Milliseconds
Number of trials	Integer count
Protocols	IPv4, IPv6, MPLS, etc.
Frame Size	Octets
Port Media	Ethernet, Gigabit Ethernet (GbE), Packet over SONET (POS), etc.
Port Speed	10 Gbps, 1 Gbps, 100 Mbps, etc.
Interface Encap.	Ethernet, Ethernet VLAN, PPP, High-Level Data Link Control (HDLC), etc.

Document any configuration deltas, which are observed after the ISSU upgrade has taken effect. Note differences, which are driven by changes in the patch or release level as well as items which are aberrant changes due to software faults. In either of these cases, any unexpected behavioral changes should be analyzed and a determination made as to the impact of the change (be it functional variances or operational impacts to existing scripts or management mechanisms).

8 Security Considerations

None at this time.

9 IANA Considerations

None at this time.

10 Conclusions

None at this time.

11 References

11.1 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.

11.2 Informative References

- [3] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.
- [Fab1999] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

12 Acknowledgments

The authors wish to thank Vibin Thomas for his valued review and feedback.

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

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

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

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Authors' Addresses

Sarah Banks
Aerohive Networks
Email: sbanks@aerohive.com

Fernando Calabria
Cisco Systems
Email: fc calabri@cisco.com

Gery Czirjak
Juniper Networks
Email: gczirjak@juniper.net

Ramdas Machat
Juniper Networks
Email: rmachat@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

J. Rapp
L. Avramov
Cisco Systems, Inc
July 15, 2013

Data Center Benchmarking Methodology
draft-bmwg-dcbench-methodology-01

Abstract

The purpose of this informational document is to establish test and evaluation methodology and measurement techniques for network equipment in the data center.

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 January 16, 2014.

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
1.1. Requirements Language	5
1.2. Methodology format	5
2. Line Rate Testing	5
2.1 Objective	5
2.2 Methodology	5
2.3 Reporting Format	6
3. Buffering Testing	6
3.1 Objective	7
3.2 Methodology	7
3.3 Reporting format	9
4. Microburst Testing	10
4.1 Objective	10
4.2 Methodology	10
4.3 Reporting Format	10
5. Head of Line Blocking	11
5.1 Objective	11
5.2 Methodology	11
5.3 Reporting Format	12
6. Incast Stateful and Stateless Traffic	13
6.1 Objective	13
6.2 Methodology	13
6.3 Reporting Format	14
7. References	14
7.1. Normative References	15
7.2. Informative References	15
7.3. URL References	15
Authors' Addresses	15

1. Introduction

Traffic patterns in the data center are not uniform and are constantly changing. They are dictated by the nature and variety of applications utilized in the data center. It can be largely east-west traffic flows in one data center and north-south in another, while some may combine both. Traffic patterns can be bursty in nature and contain many-to-one, many-to-many, or one-to-many flows. Each flow may also be small and latency sensitive or large and throughput sensitive while containing a mix of UDP and TCP traffic. All of which can coexist in a single cluster and flow through a single network device all at the same time. Benchmarking of network devices have long used RFC1242, RFC2432, RFC2544, RFC2889 and RFC3918. These benchmarks have largely been focused around various latency attributes and max throughput of the Device Under Test [DUT] being

benchmarked. These standards are good at measuring theoretical max throughput, forwarding rates and latency under testing conditions however, they do not represent real traffic patterns that may affect these networking devices.

The following provides a methodology for benchmarking Data Center DUT including congestion scenarios, switch buffer analysis, microburst, head of line blocking, while also using a wide mix of traffic conditions.

1.1. 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 [6].

1.2. Methodology format

The format used for each section of this document is the following:

-Objective

-Methodology

-Reporting Format

MUST: minimum test for the scenario described

SHOULD: recommended test for the scenario described

MAY: ideal test for the scenario described

2. Line Rate Testing

2.1 Objective

Provide at maximum rate test for the performance values for throughput, latency and jitter. It is meant to provide the tests to run and methodology to verify that a DUT is capable of forwarding packets at line rate under non-congested conditions.

2.2 Methodology

A traffic generator MUST be connected to all ports on the DUT. Two tests MUST be conducted: a port-pair test [RFC 2544/3918 compliant] and also in a full mesh type of DUT test [RFC 2889/3918 compliant].

For all tests, the percentage of traffic per port capacity sent MUST be 99.98% at most, with no PPM adjustment to ensure stressing the DUT in worst case conditions. Tests results at a lower rate MAY be provided for better understanding of performance increase in terms of latency and jitter when the rate is lower than 99.98%. The receiving rate of the traffic needs to be captured during this test in % of line rate.

The test MUST provide the latency values for minimum, average and maximum, for the exact same iteration of the test.

The test MUST provide the jitter values for minimum, average and maximum, for the exact same iteration of the test.

2.3 Reporting Format

The report MUST include:

- physical layer calibration information as defined into (Placeholder for definitions draft)

- number of ports used

- reading for throughput received in percentage of bandwidth, while sending 99.98% of port capacity on each port, across packet size from 64 byte all the way to 9216. As guidance, an increment of 64 byte packet size between each iteration being ideal, a 256 byte and 512 bytes being also often time used, the most common packets sizes order for the report is: 64b,128b,256b,512b,1024b,1518,9016b.

- throughput needs to be expressed in % of line rate

- for packet drops, they MUST be expressed in packet count value and SHOULD be expressed in % of line rate

- for latency and jitter, values expressed in unit of time [usually microsecond or nanosecond] reading across packet size from 64 bytes to 9216 bytes

- for latency and jitter, provide minimum, average and maximum values. if different iterations are done to gather the minimum, average and maximum, it SHOULD be specified in the report along with a justification on why the information could not have been gathered at the same test iteration

- for jitter, a histogram describing the population of packets measured per latency or latency buckets is RECOMMENDED

- The tests for throughput, latency and jitter MAY be conducted as individual independent events, with proper documentation in the report but SHOULD be conducted at the same time.

3. Buffering Testing

3.1 Objective

To measure the size of the buffer of a DUT under all conditions. Buffer architectures between multiple DUTs can differ and include egress buffering, shared egress buffering switch-on-chip [SoC], ingress buffering or a combination. The test methodology covers the buffer measurement regardless of buffer architecture used in the DUT.

3.2 Methodology

A traffic generator MUST be connected to all ports on the DUT.

The methodology for measuring buffering for a data-center switch is based on using known congestion of known fixed packet size along with maximum latency value measurements. The maximum latency will increase until the first packet drop occurs. At this point, the maximum latency value will remain constant. This is the point of inflexion of this maximum latency change to a constant value. There MUST be multiple ingress ports receiving known amount of frames at a known fixed size, destined for the same egress port in order to create a known congestion event. The total amount of packets sent from the oversubscribed port minus one, multiplied by the packet size represents the maximum port buffer size at the measured inflexion point.

1) Measure the highest buffer efficiency

First iteration: ingress port 1 sending line rate to egress port 2, while port 3 sending a known low amount of over subscription traffic (1% recommended) with a packet size of 64 bytes to egress port 2. Measure the buffer size value of the number of frames sent from the port sending the oversubscribed traffic up to the inflexion point multiplied by the frame size.

Second iteration: ingress port 1 sending line rate to egress port 2, while port 3 sending a known low amount of over subscription traffic (1% recommended) with same packet size 65 bytes to egress port 2. Measure the buffer size value of the number of frames sent from the port sending the oversubscribed traffic up to the inflexion point multiplied by the frame size.

Last iteration: ingress port 1 sending line rate to egress port 2, while port 3 sending a known low amount of over subscription traffic (1% recommended) with same packet size B bytes to egress port 2. Measure the buffer size value of the number of frames sent from the port sending the oversubscribed traffic up to the inflexion point multiplied by the frame size..

When the B value is found to provide the highest buffer size, this is the highest buffer efficiency

2) Measure maximum port buffer size

At fixed packet size B determined in 3.2.1, for a fixed default COS value of 0 and for unicast traffic proceed with the following:

First iteration: ingress port 1 sending line rate to egress port 2, while port 3 sending a known low amount of over subscription traffic (1% recommended) with same packet size to the egress port 2. Measure the buffer size value by multiplying the number of extra frames sent by the frame size.

Second iteration: ingress port 2 sending line rate to egress port 3, while port 4 sending a known low amount of over subscription traffic (1% recommended) with same packet size to the egress port 3. Measure the buffer size value by multiplying the number of extra frames sent by the frame size.

Last iteration: ingress port N-2 sending line rate traffic to egress port N-1, while port N sending a known low amount of over subscription traffic (1% recommended) with same packet size to the egress port N. Measure the buffer size value by multiplying the number of extra frames sent by the frame size.

This test series MAY be repeated using all different COS values of traffic and then using Multicast type of traffic.

3) Measure maximum port pair buffer sizes

First iteration: ingress port 1 sending line rate to egress port 2; ingress port 3 sending line rate to egress port 4 etc. Ingress port N-1 and N will respectively over subscribe at 1% of line rate egress port 2 and port 3. Measure the buffer size value by multiplying the number of extra frames sent by the frame size for each egress port.

Second iteration: ingress port 1 sending line rate to egress port 2; ingress port 3 sending line rate to egress port 4 etc. Ingress port N-1 and N will respectively over subscribe at 1% of line rate egress port 4 and port 5. Measure the buffer size value by multiplying the number of extra frames sent by the frame size for each egress port.

Last iteration: ingress port 1 sending line rate to egress port 2; ingress port 3 sending line rate to egress port 4 etc. Ingress port N-1 and N will respectively over subscribe at 1% of line rate egress port N-3 and port N-2. Measure the buffer size value by multiplying the number of extra frames sent by the frame size for each egress

port.

This test series MAY be repeated using all different COS values of traffic and then using Multicast type of traffic.

4) Measure maximum DUT buffer size with many to one ports

First iteration: ingress port 1,2,... N-1 sending each $[(N-1)/(\text{port capacity}) * 99.98]$ % of line rate per port to the N egress port.

Second iteration: ingress port 2,... N sending each $[(N-1)/(\text{port capacity}) * 99.98]$ % of line rate per port to the 1 egress port.

Last iteration: ingress port N,1,2,...N-2 sending each $[(N-1)/(\text{port capacity}) * 99.98]$ % of line rate per port to the N-1 egress port.

This test series MAY be repeated using all different COS values of traffic and then using Multicast type of traffic.

Unicast traffic and then Multicast traffic SHOULD be used in order to determine the proportion of buffer for documented selection of tests. Also the COS value for the packets SHOULD be provided for each test iteration as the buffer allocation size MAY differ per COS value. It is RECOMMENDED that the ingress and egress ports are varied in a random, but documented fashion in multiple tests to measure the buffer size for each port of the DUT.

3.3 Reporting format

The report MUST include:

- The packet size used for the most efficient buffer used, along with COS value
- The maximum port buffer size for each port
- The maximum DUT buffer size
- The packet size used in the test
- The amount of over subscription if different than 1%
- The number of ingress and egress ports along with their location on the DUT.

4 Microburst Testing

4.1 Objective

To find the maximum amount of packet bursts a DUT can sustain under various configurations.

4.2 Methodology

A traffic generator MUST be connected to all ports on the DUT. In order to cause congestion, two or more ingress ports MUST burst packets destined for the same egress port. The simplest of the setups would be two ingress ports and one egress port (2-to-1).

The burst MUST be measure with an intensity of 100%, meaning the burst of packets will be sent with a minimum inter-packet gap. The amount of packet contained in the burst will be variable and increase until there is a non-zero packet loss measured. The aggregate amount of packets from all the senders will be used to calculate the maximum amount of microburst the DUT can sustain.

It is RECOMMENDED that the ingress and egress ports are varied in multiple tests to measure the maximum microburst capacity.

The intensity of a microburst MAY be varied in order to obtain the microburst capacity at various ingress rates.

It is RECOMMENDED that all ports on the DUT will be tested simultaneously and in various configurations in order to understand all the combinations of ingress ports, egress ports and intensities.

An example would be:

First Iteration: N-1 Ingress ports sending to 1 Egress Ports

Second Iterations: N-2 Ingress ports sending to 2 Egress Ports

Last Iterations: 2 Ingress ports sending to N-2 Egress Ports

4.3 Reporting Format

The report MUST include:

- The maximum value of packets received per ingress port with the maximum burst size obtained with zero packet loss
- The packet size used in the test

- The number of ingress and egress ports along with their location on the DUT

5. Head of Line Blocking

5.1 Objective

Head-of-line blocking (HOL blocking) is a performance-limiting phenomenon that occurs when packets are held-up by the first packet ahead waiting to be transmitted to a different output port. This is defined in RFC 2889 section 5.5. Congestion Control. This section expands on RFC 2889 in the context of Data Center Benchmarking

The objective of this test is to understand the DUT behavior under head of line blocking scenario and measure the packet loss.

5.2 Methodology

In order to cause congestion, head of line blocking, groups of four ports are used. A group has 2 ingress and 2 egress ports. The first ingress port MUST have two flows configured each going to a different egress port. The second ingress port will congest the second egress port by sending line rate. The goal is to measure if there is loss for the first egress port which is not not oversubscribed.

A traffic generator MUST be connected to at least eight ports on the DUT and SHOULD be connected using all the DUT ports.

1) Measure two groups with eight DUT ports

First iteration: measure the packet loss for two groups with consecutive ports

The first group is composed of: ingress port 1 is sending 50% of traffic to egress port 3 and ingress port 1 is sending 50% of traffic to egress port 4. Ingress port 2 is sending line rate to egress port 4. Measure the amount of traffic loss for the traffic from ingress port 1 to egress port 3.

The second group is composed of: ingress port 5 is sending 50% of traffic to egress port 7 and ingress port 5 is sending 50% of traffic to egress port 8. Ingress port 6 is sending line rate to egress port 8. Measure the amount of traffic loss for the traffic from ingress port 5 to egress port 7.

Second iteration: repeat the first iteration by shifting all the ports from N to N+1

the first group is composed of: ingress port 2 is sending 50% of traffic to egress port 4 and ingress port 2 is sending 50% of traffic to egress port 5. Ingress port 3 is sending line rate to egress port 5. Measure the amount of traffic loss for the traffic from ingress port 2 to egress port 4.

the second group is composed of: ingress port 6 is sending 50% of traffic to egress port 8 and ingress port 6 is sending 50% of traffic to egress port 9. Ingress port 7 is sending line rate to egress port 9. Measure the amount of traffic loss for the traffic from ingress port 6 to egress port 8.

Last iteration: when the first port of the first group is connected on the last DUT port and the last port of the second group is connected to the seventh port of the DUT

Measure the amount of traffic loss for the traffic from ingress port N to egress port 2 and from ingress port 4 to egress port 6.

2) Measure with N/4 groups with N DUT ports

First iteration: Expand to fully utilize all the DUT ports in increments of four. Repeat the methodology of 1) with all the group of ports possible to achieve on the device and measure for each port group the amount of traffic loss.

Second iteration: Shift by +1 the start of each consecutive ports of groups

Last iteration: Shift by N-1 the start of each consecutive ports of groups and measure the traffic loss for each port group.

5.3 Reporting Format

For each test the report MUST include:

- The port configuration including the number and location of ingress and egress ports located on the DUT
- If HOLB was observed
- Percent of traffic loss

6. Incast Stateful and Stateless Traffic

6.1 Objective

The objective of this test is to measure the effect of TCP Goodput and latency with a mix of large and small flows. The test is designed to simulate a mixed environment of stateful flows that require high rates of goodput and stateless flows that require low latency.

6.2 Methodology

In order to simulate the effects of stateless and stateful traffic on the DUT there MUST be multiple ingress ports receiving traffic destined for the same egress port. There also MAY be a mix of stateful and stateless traffic arriving on a single ingress port. The simplest setup would be 2 ingress ports receiving traffic destined to the same egress port.

One ingress port MUST be maintaining a TCP connection through the ingress port to a receiver connected to an egress port. Traffic in the TCP stream MUST be sent at the maximum rate allowed by the traffic generator. At the same time the TCP traffic is flowing through the DUT the stateless traffic is sent destined to a receiver on the same egress port. The stateless traffic MUST be a microburst of 100% intensity.

It is RECOMMENDED that the ingress and egress ports are varied in multiple tests to measure the maximum microburst capacity.

The intensity of a microburst MAY be varied in order to obtain the microburst capacity at various ingress rates.

It is RECOMMENDED that all ports on the DUT be used in the test.

For example:

Stateful Traffic port variation:

During Iterations number of Egress ports MAY vary as well.

First Iteration: 1 Ingress port receiving stateful TCP traffic and 1 Ingress port receiving stateless traffic destined to 1 Egress Ports

Second Iteration: 2 Ingress port receiving stateful TCP traffic and 1 Ingress port receiving stateless traffic destined to 1 Egress Ports

Last Iteration: N-2 Ingress port receiving stateful TCP traffic and 1 Ingress port receiving stateless traffic destined to 1 Egress Ports

Stateless Traffic port variation:

During Iterations number of Egress ports MAY vary as well. First Iteration: 1 Ingress port receiving stateful TCP traffic and 1 Ingress port receiving stateless traffic destined to 1 Egress Ports

Second Iteration: 1 Ingress port receiving stateful TCP traffic and 2 Ingress port receiving stateless traffic destined to 1 Egress Ports

Last Iteration: 1 Ingress port receiving stateful TCP traffic and N-2 Ingress port receiving stateless traffic destined to 1 Egress Ports

6.3 Reporting Format

The report MUST include the following:

- Number of ingress and egress ports along with designation of stateful or stateless.
- TCP flow goodput
- Stateless flow latency

7. References

7.1. Normative References

- [1] Bradner, S. "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, July 1991.
- [2] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

7.2. Informative References

- [3] Mandeville R. and Perser J., "Benchmarking Methodology for LAN Switching Devices", RFC 2889, August 2000.
- [4] Stopp D. and Hickman B., "Methodology for IP Multicast Benchmarking", BCP 26, RFC 3918, October 2004.

7.3. URL References

- [5] Yanpei Chen, Rean Griffith, Junda Liu, Randy H. Katz, Anthony D. Joseph, "Understanding TCP Incast Throughput Collapse in Datacenter Networks",
<http://www.eecs.berkeley.edu/~ychen2/professional/TCPIncastWREN2009.pdf>

Authors' Addresses

Jacob Rapp
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States
Phone: +1 408 853 2970
Email: jarapp@cisco.com

Lucien Avramov
Cisco Systems
170 West Tasman drive
San Jose, CA 95134
United States
Phone: +1 408 526 7686
Email: lavramov@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2013

W.J. Cervený
Arbor Networks
March 11, 2013

Benchmarking Neighbor Discovery Problems
draft-cervený-bmwg-ipv6-nd-00

Abstract

This document is a benchmarking instantiation of RFC 6583: "Operational Neighbor Discovery Problems". 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 September 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. Introduction	3
2. Terminology	3
3. Test Set-up	3
3.1. Device Under Test (DUT)	3
3.2. Test Network	3
4. Modifiers (variables)	4
4.1. Frequency of NDP triggering events	4
4.2. Prefix Length	5
4.3. Duration of Test	5
4.4. Packet Size	5
4.5. Packet Type	5
4.6. Packet Addressing	5
4.7. Testing of Mitigating Options	5
4.8. Attack where node in target network responds to all neighbor solicitations	6
5. Exclusions	6
6. Measurements	6
6.1. Round-trip time across DUT	6
6.2. Rate DUT adds a valid node in the target network to its neighbor cache	6
6.3. Adherence to prioritization of NDP activity prioritization	7
6.4. DUT CPU utilization	7
6.5. Rate DUT forwards packets	7
6.6. Rate DUT responds to neighbor solicitations for its own address	7
6.7. Impact on unaffected interfaces/subnets	8
6.8. Maximum number of entries in the DUT's neighbor cache .	8
7. Measurement Interval	8
8. DUT initialization	8
9. General Test Procedure	8
10. Other Potential Testing Scenarios	9
10.1. Exhaustion of Address Tables (NCE) in Intermediate Nodes	9
10.2. Link-local network attack	9
11. IANA Considerations	9
12. Security Considerations	9
13. Acknowledgements	10
14. Normative References	10
Author's Address	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

Neighbor Discovery See Section 1 of RFC 4861 [RFC4861]

NDP Triggering Event An event which forces the DUT (Device Under Test) to perform a neighbor solicitation. A triggering event could be an ICMPv6 echo request, but could also be any other packets which require discovering the MAC address of existing and non-existing nodes on an IPv6 subnet.

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

Target Network The network for which the scanner is targeting its scans.

Scanning Node The node which is conducting the scanning activity.

Target Network Measurement Node A node that resides on the target network, which is primarily used to measure DUT performance while the scanning activity is occurring.

Non-participating Measurement Node A node on a network directly connected to the DUT, but this node is not in the target network nor the scanner network.

3. Test Set-up

3.1. Device Under Test (DUT)

For purposes of this document, the intermediate node will be referred to as the device under test (DUT). The DUT may be any intermediate node which retains a neighbor cache. The tests in this document could also be completed with any intermediate node which maintains a list of addresses that traverse the intermediate node, although not all measurements and performance characteristics may apply.

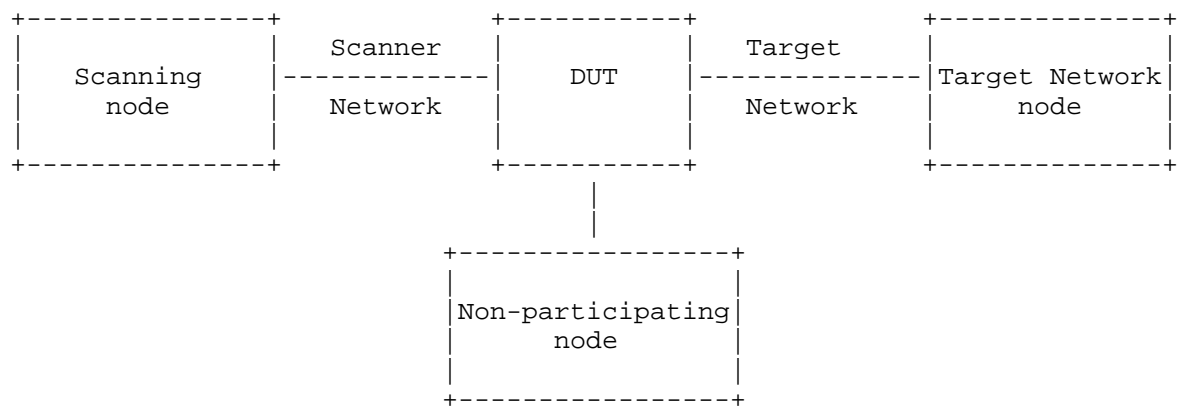
3.2. Test Network

The test network design is fairly simple. 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.

At least one node should reside on the target network to confirm some of the performance characteristics.

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



4. Modifiers (variables)

4.1. Frequency of NDP triggering events

The frequency of NDP triggering events could be as high as the maximum packet per second rate that the scanner network will support (or is rated for). However, it may not be necessary to send packets at a particularly high rate and in fact a goal of testing could be to identify if the DUT is able to withstand scans at rates which otherwise would not impact the performance of the DUT.

Optimistically, the scanning rate should be incremented until the DUT's performance begins deteriorating. Depending on the software and system being used to implement the scanning, it may be challenging to achieve a sufficient rate.

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

4.2. Prefix Length

The target network's subnet shall be 64-bits in length. It may be interesting to gauge performance when the subnet length is varied from 64-bits.

4.3. Duration of Test

The duration of the test needs to be evaluated

4.4. Packet Size

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

4.5. Packet Type

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. Since it is not intended that responses be evoked from the target network node, such packets aren't necessary.

The hop limit for the scanning packets should be set to 2, to reduce the likelihood that scanning packets would escape the test network.

4.6. Packet Addressing

The destination address for the packet should be an address within the target network. While each packet sent should have a unique destination address in the destination network, it isn't clear if it matters what the sequence of addresses is. For purposes of thoroughness, it may be desirable to send each packet with a random address within the target network's address space.

The source address for the packet may be the same for all scanning packets. However, it may be interesting to vary the source address during the scanning activity

4.7. Testing of Mitigating Options

It may be desirable to perform some tests in the presence of mitigating techniques described in RFC 6583 [RFC6583]

4.8. Attack where node in target network responds to all neighbor solicitations

[Open Question: Is this an interesting condition, where a device on the network responds affirmatively to all incoming NDP requests?? Are there any non-malicious cases where this could happen?]

5. Exclusions

This benchmarking test is not intended to test DUT behavior in the presence of malformed packets, such as packets which do not confirm to designs consistent with IETF standards.

6. Measurements

6.1. Round-trip time across DUT

This consists of pinging the target network measurement node from a non-participating measurement node and recording reported round-trip time. This measurement should be conducted with an address not yet present in the DUT's neighbor cache. This measurement is included because it is perhaps the easiest to conduct and capture.

6.2. Rate DUT adds a valid node in the target network to its neighbor cache

There are three distinct time elements associated with this measurement:

1. The difference in time for which the DUT receives the packet which must be forwarded to a node in the target network not yet listed in the neighbor cache and the time the DUT sends a neighbor solicitation.
2. The difference in time between when the target network measurement node receives the neighbor solicitation and the time the target network measurement node responds with a neighbor advertisement. This time is outside the control of the DUT and measurements should account for this time if it is significant.
3. The difference in time from which the DUT receives the packet to the time for which the DUT adds the neighbor in its neighbor cache.

The first time element may be measurable via a device which can observe packets on both the scanner network and the target network. The second time element may be measured by monitoring the target network and observing the specific neighbor solicitation for the node

and the node's solicited[Is this the right term?] neighbor advertisement.

Of the above time elements, the third is perhaps the hardest to measure for times smaller than a few seconds.

A challenge with this measurement is to conduct it where the target network node has an address that is not in the DUT's neighbor cache in any state (such as "INCOMPLETE"). As tested with a router, the router's "clear neighbor cache" command did not always flush the target network node's neighbor entry. One method of implementing this may be to configure the target network node with sufficient addresses for a unique NDP request per test interval.

6.3. Adherence to prioritization of NDP activity prioritization

As discussed in RFC 6583 [RFC6583], this measurement would require confirming that a set prioritization is adhered to. [Insert more text here.]

6.4. DUT CPU utilization

Measured in percent utilization, captured via a non-intrusive query of the DUT.

6.5. Rate DUT forwards packets

This measures the impact that the scan may have on the DUT's ability to forward packets. The measurement should be documented in packets per seconds (pps) or (bps). However, if the DUT handles NDP in the "management plane" and packets are forwarded in a separate "forwarding plane", the scanning tests described in this document may not have any impact on the DUT's ability to forward packets.

It may be beneficial to conduct two RFC 5180 [RFC5180] style throughput tests even if it is assumed that scanning activity won't have any bearing on the DUT's packet forwarding capabilities:

1. Baseline test without any scanning activity.
2. Test while worse-case scanning activity is occurring.

6.6. Rate DUT responds to neighbor solicitations for its own address

This is the difference in time from when a node on the target network sends a neighbor solicitation for the DUT's MAC address and when the DUT responds with a neighbor advertisement in response to the neighbor solicitation. This can be determined by observing the

target network and measuring the difference in time (in milliseconds) between when the neighbor solicitation leaves the target network measurement node and when the solicited neighbor advertisement is returned from the DUT.

6.7. Impact on unaffected interfaces/subnets

This measurement would require having a node on a network directly connected to the DUT, but not on either the scanner network or target network. Although not itemized, this measurement could consist of any combination of measurements which are conducted relating to the target network.

6.8. Maximum number of entries in the DUT's neighbor cache

This measurement confirms how many entries can effectively reside in the DUT's neighbor cache. This measurement would support or refute any value documented by the DUT manufacturer. [Need to describe how this is done.]

7. Measurement Interval

To be determined.

8. DUT initialization

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

9. General Test Procedure

This test can be completed with publicly available scanning software. The methodology to implement this scan is fairly straightforward and could be implemented using open-source network scripting tools.

The algorithm for such a scanner could be as simple as:

```
Dest_address = <ip prefix>::1000
```

```
While True:
```

```
Send(ICMPv6(dst=Dest_address))
```

```
Dest_address = Dest_address + 1
```

As described in [RFC6583], four instances of a scanner on a single computer was able to impact the performance of high-end routers. If multiple scanner instances are used, the starting address should be in different "regions" of the subnet.

Some existing software for completing network scans is discussed in [RFC6583], although other applications may exist.

Although not tested, commercial network testing solutions may be effectively implemented and may provide desired throughput.

10. Other Potential Testing Scenarios

10.1. Exhaustion of Address Tables (NCE) in Intermediate Nodes

[Question: Where a large number of addresses are being scanned for, would there be an impact on intermediate nodes, such as firewalls?]

10.2. Link-local network attack

In this attack, a node in the subnet simulates a condition where it is sending packets to every address in the subnet and where the destination MAC address is the DUT[Is this an allowed scenario?]. In this scenario, it "could" be possible to send neighbor solicitation messages to every link local address via the default gateway.

11. IANA Considerations

This document makes no request of IANA.

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

12. 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.

13. Acknowledgements

14. 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.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, May 2008.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.

Author's Address

Bill Cervený
Arbor Networks

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.

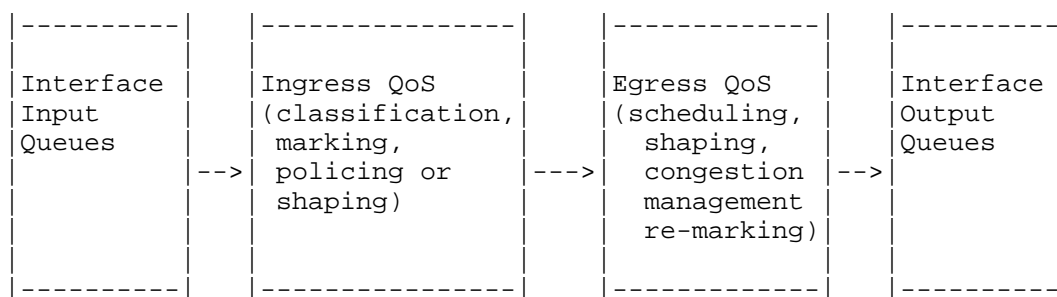


Figure 1: Generic Model of Traffic Management capabilities within a network device

(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

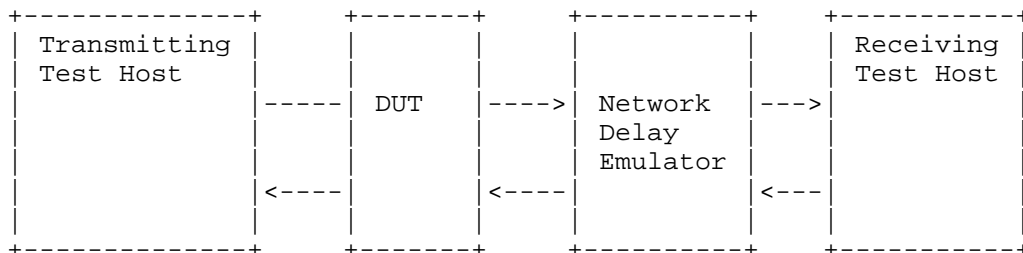
QoS: Quality of Service

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.

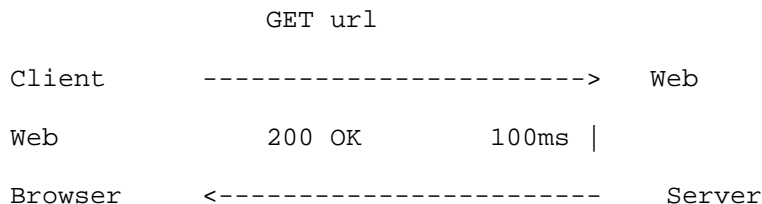
The test host must not only support bulk TCP transfer application traffic but this chatty traffic since the both stress traffic management techniques in very different ways. This is due to the non-uniform, bursty nature of chatty applications versus the relatively uniform nature of bulk transfers (the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

$RTT(\text{min}) = QL * 8 / BB$ 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

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: December 6, 2013

J. Rapp
L. Avramov
Cisco Systems, Inc
June 4, 2013

Definitions and Metrics for Data Center Benchmarking
draft-dcbench-def-00

Abstract

The purpose of this informational document is to establish definitions, discussion and measurement techniques for data center benchmarking. Also, it is to introduce new terminologies applicable to data center performance evaluations. The purpose of this document is not to define the test methodology, but rather establish the important concepts when one is interested in benchmarking network equipment in the data center.

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 6, 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
1.1. Requirements Language	4
1.2. Definition format	4
2. Latency	4
2.1. Definition	4
2.2 Discussion	5
2.3 Measurement	5
3 Jitter	5
3.1 Definition	5
3.2 Discussion	6
3.3 Measurement	6
4 Physical Layer Calibration	6
4.1 Definition	6
4.2 Discussion	7
4.3 Measurement	7
5 Line rate	7
5.1 Definition	7
5.2 Discussion	8
5.3 Measurement	9
6 Buffering	9
6.1 Buffer	9
6.1.1 Definition	9
6.1.2 Discussion	11
6.1.3 Measurement	11
6.2 Incast	11
6.2.1 Definition	11
6.2.2 Discussion	12
6.2.3 Measurement	12
7 Application Throughput: Data Center Goodput	12
7.1. Definition	12
7.2. Discussion	13
7.3. Measurement	13
8. References	13
3.1. Normative References	14
3.2. Informative References	14
3.3. URL References	14
3.4. Acknowledgments	14
Authors' Addresses	14

1. Introduction

Traffic patterns in the data center are not uniform and are contently changing. They are dictated by the nature and variety of applications utilized in the data center. It can be largely east-west traffic flows in one data center and north-south in another, while some may combine both. Traffic patterns can be bursty in nature and contain many-to-one, many-to-many, or one-to-many flows. Each flow may also be small and latency sensitive or large and throughput sensitive while containing a mix of UDP and TCP traffic. All of which can coexist in a single cluster and flow through a single network device all at the same time. Benchmarking of network devices have long used RFC1242, RFC2432, RFC2544, RFC2889 and RFC3918. These benchmarks have largely been focused around various latency attributes and max throughput of the Device Under Test being benchmarked. These standards are good at measuring theoretical max throughput, forwarding rates and latency under testing conditions, but to not represent real traffic patterns that may affect these networking devices.

The following defines a set of definitions, metrics and terminologies including congestion scenarios, switch buffer analysis and redefines basic definitions in order to represent a wide mix of traffic conditions.

1.1. 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 [6].

1.2. Definition format

Term to be defined. (e.g., Latency)

Definition: The specific definition for the term.

Discussion: A brief discussion about the term, it's application and any restrictions on measurement procedures.

Measurement: Methodology for the measure and units used to report measurements of this term, if applicable.

2. Latency

2.1. Definition

Latency is a the amount of time it takes a frame to transit the DUT.

Latency can be measured with the following methods, irrespectively of the type of switching device (bit forwarding aka cut-through or store forward type of device)

FILO (First In Last Out) The time interval starting when the end of the first bit of the input frame reaches the input port and ending when the last bit of the output frame is seen on the output port

FIFO (First In First Out) The time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port

LILO (Last In Last Out) The time interval starting when the last bit of the input frame reaches the input port and the last bit of the output frame is seen on the output port

LIFO (Last In First Out) The time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port.

This definition replaces the previous definition of Latency defined in RFC 1242, section 3.8 and is quoted here:

For store and forward devices: The time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port.

For bit forwarding devices: The time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port.

2.2 Discussion

FILO is the most important measuring method. Any type of switches MUST be measured with the FILO mechanism: FILO will include the latency of the switch and the latency of the frame as well as the serialization delay. It is a picture of the 'whole' latency going through the DUT. For applications, which are latency sensitive and can function with initial bytes of the frame, FIFO MAY be an additional type of measuring to supplement FILO.

LIFO mechanism can be used with store forward type of switches but not with cut-through type of switches, as it will provide negative latency values for larger packet sizes. Therefore this mechanism MUST NOT be used when comparing latencies of two different DUTs.

2.3 Measurement

The measuring methods to use for benchmarking purposes are as follow:

- 1) FILO MUST be used as a measuring method, as this will include the latency of the packet; and today the application commonly need to read the whole packet to process the information and take an action.
- 2) FIFO MAY be used for certain applications able to proceed data as the first bits arrive (FPGA for example)
- 3) LIFO MUST not be used, because it subtracts the latency of the packet; unlike all the other methods.

3 Jitter

3.1 Definition

The definition of Jitter is covered extensively in RFC 3393. This definition is not meant to replace that definition, but it is meant

to provide guidance of use for data center network devices.

The use of Jitter is in according with the variation delay definition from RFC 3393:

The second meaning has to do with the variation of a metric (e.g., delay) with respect to some reference metric (e.g., average delay or minimum delay). This meaning is frequently used by computer scientists and frequently (but not always) refers to variation in delay.

3.2 Discussion

Jitter can be measured in different scenarios:-packet to packet delay variation-delta between min and max packet delay variation for all packets sent.

3.3 Measurement

The jitter MUST be measured when sending packets of the same size. Jitter MUST be measured as packet to packet delay variation and delta between min and max packet delay variation of all packets sent. A histogram MAY be provided as a population of packets measured per latency or latency buckets.

4 Physical Layer Calibration

4.1 Definition

The calibration of the physical layer consists of defining and measuring the latency of the physical devices used to perform test on the DUT.

It includes the list of all physical layer components used as listed here after:

- type of device used to generate traffic / measure traffic
- type of line cards used on the traffic generator
- type of transceivers on traffic generator
- type of transceivers on DUT
- type of cables
- length of cables

- software name, and version of traffic generator and DUT
- list of enabled features on DUT MAY be provided and is recommended [especially the control plane protocols such as LLDP, Spanning-Tree etc.]. A comprehensive configuration file MAY be provided to this effect.

4.2 Discussion

Physical layer calibration is part of the end to end latency, which should be taken into acknowledgment while evaluating the DUT. Small variations of the physical components of the test may impact the latency being measure so they MUST be described when presenting results.

4.3 Measurement

It is RECOMMENDED to use all cables of : the same type, the same length, when possible using the same vendor. It is a MUST to document the cables specifications on section [4.1s] along with the test results. The test report MUST specify if the cable latency has been removed from the test measures or not. The accuracy of the traffic generator measure MUST be provided [this is usually a value in the 20ns range for current test equipments].

5 Line rate

5.1 Definition

The transmit timing, or maximum transmitted data rate is controlled by the "transmit clock" in the DUT. The receive timing (maximum ingress data rate) is derived from the transmit clock of the connected interface.

The line rate or physical layer frame rate is the maximum capacity to send frames of a specific size at the transmit clock frequency of the DUT.

The frequency ("clock rate") of the transmit clock in any two connected interfaces will never be precisely the same, therefore a tolerance is needed, this will be expressed by Parts Per Million (PPM) value. The IEEE standards allow a specific +/- variance in the transmit clock rate, and Ethernet is designed to allow for small, normal variations between the two clock rates. This results in a tolerance of the line rate value when traffic is generated from a testing equipment to a DUT.

5.2 Discussion

For a transmit clock source, most Ethernet switches use "clock modules" (also called "oscillator modules") that are sealed, internally temperature-compensated, and very accurate. The output frequency of these modules is not adjustable because it is not necessary. Many test sets, however, offer a software-controlled adjustment of the transmit clock rate, which should be used to compensate the test equipment to not send more than line rate of the DUT.

To allow for the minor variations typically found in the clock rate of commercially-available clock modules and other crystal-based oscillators, Ethernet standards specify the maximum transmit clock rate variation to be not more than ± 100 PPM (parts per million) from a calculated center frequency. Therefore a DUT must be able to accept frames at a rate within ± 100 PPM to comply with the standards.

Very few clock circuits are precisely ± 0.0 PPM because:

- 1.The Ethernet standards allow a maximum of ± 100 PPM (parts per million) variance over time. Therefore it is normal for the frequency of the oscillator circuits to experience variation over time and over a wide temperature range, among external factors.
- 2.The crystals or clock modules, usually have a specific \pm PPM variance that is significantly better than ± 100 PPM. Often times this is ± 30 PPM or better in order to be considered a "certification instrument".

When testing an Ethernet switch throughput at "line rate", any specific switch will have a clock rate variance. If a test set is running ± 1 PPM faster than a switch under test, and a sustained line rate test is performed, a gradual increase in latency and eventually packet drops as buffers fill and overflow in the switch can be observed. Depending on how much clock variance there is between the two connected systems, the effect may be seen after the traffic stream has been running for a few hundred microseconds, a few milliseconds, or seconds. The same low latency and no-packet-loss can be demonstrated by setting the test set link occupancy to slightly less than 100 percent link occupancy. Typically 99 percent link occupancy produces excellent low-latency and no packet loss. No Ethernet switch or router will have a transmit clock rate of exactly ± 0.0 PPM. Very few (if any) test sets have a clock rate that is precisely ± 0.0 PPM.

Test set equipment manufacturers are well-aware of the standards, and

allows a software-controlled +/- 100 PPM "offset" (clock-rate adjustment) to compensate for normal variations in the clock speed of "devices under test". This offset adjustment allows engineers to determine the approximate speed the connected device is operating, and verify that it is within parameters allowed by standards.

5.3 Measurement

"Line Rate" CAN be measured in terms of "Frame Rate":

Frame Rate = Transmit-Clock-Frequency / (Frame-Length*8 + Minimum_Gap + Preamble + Start-Frame Delimiter)

Example for 1 GB Ethernet speed with 64-byte frames: Frame Rate = 1,000,000,000 / (64*8 + 96 + 56 + 8) Frame Rate = 1,000,000,000 / 672
Frame Rate = 1,488,095.2 frames per second.

Considering the allowance of +/- 100 PPM, a switch may "legally" transmit traffic at a frame rate between 1,487,946.4 FPS and 1,488,244 FPS. Each 1 PPM variation in clock rate will translate to a 1.488 frame-per-second frame rate increase or decrease.

In a production network, it is very unlikely to see precise line rate over a very brief period. There is no observable difference between dropping packets at 99% of line rate and 100% of line rate.

-Line rate CAN be measured at 100% of line rate with a -100PPM adjustment.

-Line rate SHOULD be measured at 99.98% with 0 PPM adjustment.

6 Buffering

6.1 Buffer

6.1.1 Definition

Buffer Size: the term buffer size, represents the total amount of frame buffering memory available on a DUT. This size is expressed in Byte; KB (kilobytes), MB (megabytes) or GB (gigabyte). When the buffer size is expressed it SHOULD be defined by a size metric defined above. When the buffer size is expressed, an indication of the frame MTU used for that measurement is also necessary as well as the cos or dscp value set; as often times the buffers are carved by quality of service implementation.

Example: Buffer Size of DUT when sending 1518 bytes frames is 18 Mb.

Port Buffer Size: the port buffer size is the amount of buffer a single ingress port, egress port or combination of ingress and egress buffering location for a single port. The reason of mentioning the three locations for the port buffer is, that the DUT buffering scheme can be unknown or untested, and therefore the indication of where the buffer is located helps understand the buffer architecture and therefore the total buffer size. The Port Buffer Size is an informational value that MAY be provided from the DUT vendor. It is not a value that is tested by benchmarking. Benchmarking will be done using the Maximum Port Buffer Size or Maximum Buffer Size methodology.

Maximum Port Buffer Size: this is in most cases the same as the Port Buffer Size. In certain switch architecture called SoC (switch on chip), there is a concept of port buffer and shared buffer pool available for all ports. Maximum Port Buffer, defines the scenario of a SoC buffer, where this amount in B (byte), KB (kilobyte), MB (megabyte) or GB (gigabyte) would represent the sum of the port buffer along with the maximum value of shared buffer this given port can take. The Maximum Port Buffer Size needs to be expressed along with the frame MTU used for the measurement and the cos or dscp bit value set for the test.

Example: a DUT has been measured to have 3KB of port buffer for 1518 frame size packets and a total of 4.7 MB of maximum port buffer for 1518 frame size packets and a cos of 0.

Maximum DUT Buffer Size: this is the total size of Buffer a DUT can be measured to have. It is most likely different than the Maximum Port Buffer Size. It CAN also be different from the sum of Maximum Port Buffer Size. The Maximum Buffer Size needs to be expressed along with the frame MTU used for the measurement and along with the cos or dscp value set during the test.

Example: a DUT has been measured to have 3KB of port buffer for 1518 frame size packets and a total of 4.7 MB of maximum port buffer for 1518 frame size packets. The DUT has a Maximum Buffer Size of 18 MB at 1500 bytes and a cos of 0.

Burst: The burst is a fixed number of packets sent over a percentage of linerate of a defined port speed. The amount of frames sent are evenly distributed across the interval T. A constant C, can be defined to provide the average time between two consecutive packets evenly spaced.

Microburst: it is a burst. A microburst is when packet drops occur

when there is not sustained or noticeable congestion upon a link or device. A characterization of microburst is when the Burst is not evenly distributed over T, and is less than the constant C [C= average time between two consecutive packets evenly spaced out].

Intensity of Microburst: this is a percentage, representing the level of microburst between 1 and 100%. The higher the number the higher the microburst is. $I = [1 - [(TP2 - Tp1) + (Tp3 - Tp2) + \dots + (TpN - Tp(n-1))] / \text{Sum}(\text{packets})]] * 100$

6.1.3 Discussion

When measuring buffering on a DUT, it is important to understand what the behavior is for each port, and also for all ports as this will provide an evidence of the total amount of buffering available on the switch. The terms of buffer efficiency here helps one understand what is the optimum packet size for the buffer to be used, or what is the real volume of buffer available for a specific packet size. This section does not discuss how to conduct the test methodology, it rather explains the buffer definitions and what metrics should be provided for a comprehensive data center device buffering benchmarking.

6.1.3 Measurement

When Buffer is measured:

- the buffer size MUST be measured
- the port buffer size MAY be provided for each port
- the maximum port buffer size MUST be measured
- the maximum DUT buffer size MUST be measured
- the intensity of microburst MAY be mentioned when a microburst test is performed
- the cos or dscp value set during the test SHOULD be provided

6.2 Incast

6.2.1 Definition

The term Incast, very commonly utilized in the data center, refers to the traffic pattern of many-to-one or many-to-many conversations. Typically in the data center it would refer to many different ingress server ports (many), sending traffic to a common uplink (one), or multiple uplinks (many). This pattern is generalized for any network as many incoming ports sending traffic to one or few uplinks. It can also be found in many-to-many traffic patterns.

Synchronous arrival time: When two, or more, frames of respective sizes L1 and L2 arrive at their respective one or multiple ingress

ports, and there is an overlap of the arrival time for any of the bits on the DUT, then the frames L1 and L2 have a synchronous arrival times. This is called incast.

Asynchronous arrival time: Any condition not defined by synchronous.

Percentage of synchronization: this defines the level of overlap [amount of bits] between the frames L1,L2..Ln.

Example: two 64 bytes frames, of length L1 and L2, arrive to ingress port 1 and port 2 of the DUT. There is an overlap of 6.4 bytes between the two where L1 and L2 were at the same time on the respective ingress ports. Therefore the percentage of synchronization is 10%.

6.2.2 Discussion

In this scenario, buffers are solicited on the DUT. In a ingress buffering mechanism, the ingress port buffers would be solicited along with Virtual Output Queues, when available; whereas in an egress buffer mechanism, the egress buffer of the one outgoing port would be used.

In either cases, regardless of where the buffer memory is located on the switch architecture; the Incast creates buffer utilization.

When one or more frames having synchronous arrival times at the DUT they are considered forming an incast.

6.2.3 Measurement

It is a MUST to measure the number of ingress and egress ports. It is a MUST to have a non null percentage of synchronization, which MUST be specified.

7 Application Throughput: Data Center Goodput

7.1. Definition

In Data Center Networking, a balanced network is a function of maximal throughput 'and' minimal loss at any given time. This is defined by the Goodput. Goodput is the application-level throughput. It is measured in bytes / second. Goodput is the measurement of the actual payload of the packet being sent.

7.2. Discussion

In data center benchmarking, the Goodput is a value that SHOULD be measured. It provides a realistic idea of the usage of the available bandwidth. A goal in data center environments is to maximize the Goodput while minimizing the loss.

7.3. Measurement

When S is the total bytes received from all senders [not inclusive of packet headers or TCP headers - it's the payload] and F_t is the Finishing Time of the last sender; the Goodput G is then measured by the following formula: $G = S / F_t$ bytes per second

Example: a TCP file transfer over HTTP protocol on a 10Gb/s media. The file cannot be transferred over Ethernet as a single continuous stream. It must be broken down into individual frames of 1500 bytes when the standard MTU [Maximum Transmission Unit] is used. Each packet requires 20 bytes of IP header information and 20 bytes of TCP header information, therefore 1460 bytes are available per packet for the file transfer. Linux based systems are further limited to 1448 bytes as they also carry a 12 byte timestamp. Finally, the data is transmitted in this example over Ethernet which adds a 26 byte overhead per packet.

$G = 1460 / 1526 \times 10 \text{ Gbit/s}$ which is 9.567 Gbit/s or 1.196 Gigabytes per second.

Please note: this example does not take into consideration additional Ethernet overhead, such as the interframe gap (a minimum of 96 bit times), nor collisions (which have a variable impact, depending on the network load).

When conducting Goodput measurements please document in addition to the 4.1 section:

- the TCP Stack used
- OS Versions
- NIC firmware version and model

For example, Windows TCP stacks and different Linux versions can influence TCP based tests results.

8. References

8.1. Normative References

- [1] Bradner, S. "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, July 1991.
- [2] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

8.2. Informative References

- [3] Mandeville R. and Perser J., "Benchmarking Methodology for LAN Switching Devices", RFC 2889, August 2000.
- [4] Stopp D. and Hickman B., "Methodology for IP Multicast Benchmarking", BCP 26, RFC 3918, October 2004.

8.3. URL References

- [5] Yanpei Chen, Rean Griffith, Junda Liu, Randy H. Katz, Anthony D. Joseph, "Understanding TCP Incast Throughput Collapse in Datacenter Networks",
<http://www.eecs.berkeley.edu/~ychen2/professional/TCPIncastWREN2009.pdf>

8.4. Acknowledgments

The authors would like to thank Ian Cox and Tim Stevenson for their reviews and feedback.

Authors' Addresses

Jacob Rapp
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States
Phone: +1 408 853 2970
Email: jarapp@cisco.com

Lucien Avramov
Cisco Systems
170 West Tasman drive
San Jose, CA 95134
United States
Phone: +1 408 526 7686
Email: lavramov@cisco.com

Benchmarking Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2014

R. Papneja
Huawei Technologies
B. Parise
Cisco Systems
S. Hares
Adara Networks
D. Lee
IXIA
I. Varlashkin
Easynet Global Services
July 2013

Basic BGP Convergence Benchmarking Methodology for Data Plane
Convergence
draft-ietf-bmwg-bgp-basic-convergence-00.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 January 2, 2014.

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 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	20
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	32
8. Security Considerations	32
9. Acknowledgements	32
10. References	32
10.1. Normative References	32
10.2. Informative References	33
Authors' Addresses	33

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 IGP (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.

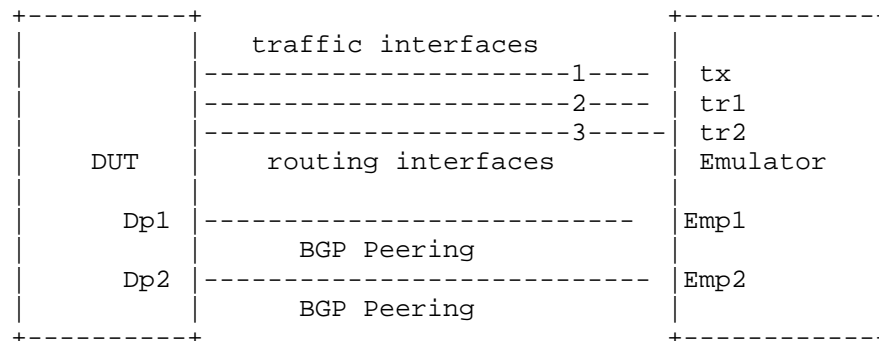


Figure 1 Basic Test Setup

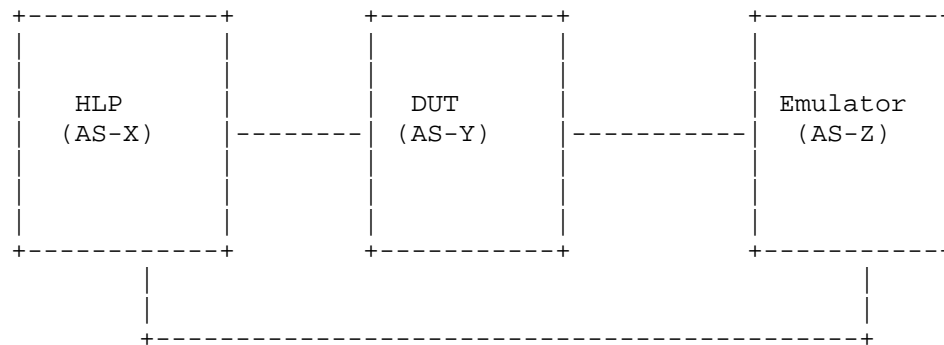


Figure 2 Three Node Setup for eBGP and iBGP Convergence

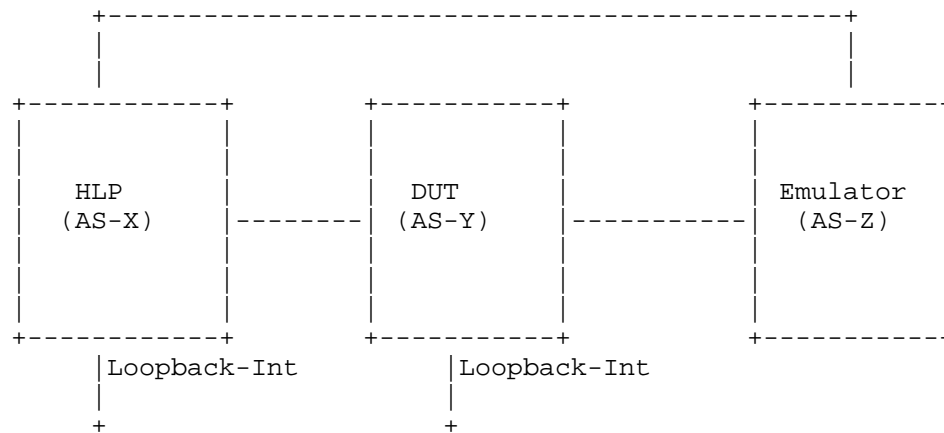


Figure 3 BGP Convergence for eBGP Multihop Scenario

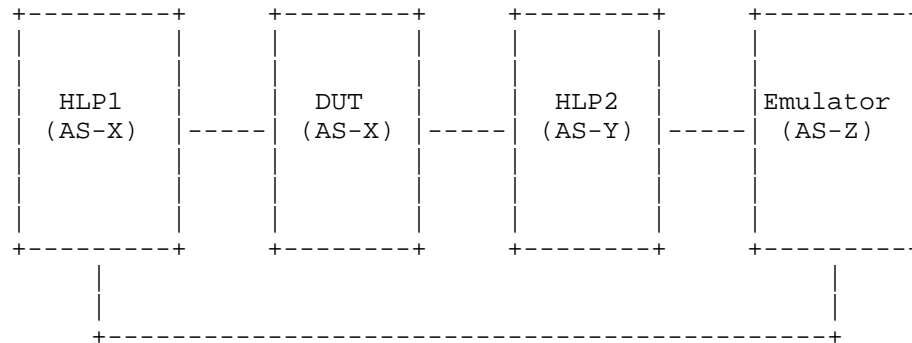


Figure 4 Four Node Setup for EBGP and IBGP Convergence

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 test case (as specified in the test).

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 advertised 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 advertising 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 $T_{up}(EMx, Rt-A)$ also named 'XMT-Rt-time(Rt-A)'.

- 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(Rt-A)'.

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

- 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 ($Rt-A$) 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 (e.g. 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 route A from the Emulator to the DUT and note the time.

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

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

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

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

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

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

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

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

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(first)-DUT-XMT-Data-Time(last)

RIB-convergence =DUT-RCV-Rt-Time(first)-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 tes.t
- I. Start the traffic from the Emulator towards the DUT targeted at a specific route (e.g. route A).
- J. Initially no traffic SHOULD be observed on the egress interface as the route A is not installed in the forwarding database of the DUT.
- K. Advertise route A from the Emulator to the DUT and note the time (Tup(EMx,RouteA) also named Route-Tx-time(Rt-A).
- L. Record the time when the route is received by the DUT. This is Tup(EMr,DUT) named Route-Rcv-time(Rt-A).
- M. Record the time when the traffic targeted towards route A is received from Egress Interface of DUT on emulator. This is Tup(EMd,DUT) named Data-Rcv-time(Rt-A)
- N. Record the time when the route A is forwarded by DUT towards the Helper node. This is Tup(EMf,DUT) also named Route-Fwd-time(Rt-A)

$$\text{FIB Convergence} = (\text{Data-Rcv-time} - \text{Route-Rcv-time})(\text{Rt-A})$$

$$\text{RIB Convergence} = (\text{Route-Fwd-time} - \text{Route-Rcv-time})(\text{Rt-A})$$

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 (e.g. route A). Initially traffic would be observed on the best egress route (Empl) 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 (e.g route A). 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 (e.g. route A). Initially no traffic would be observed on the Egress interface as the route A is not present on DUT.
 - (7) Advertise route A from the Emulator to the DUT.
 - (8) The traffic targeted towards route A is received on the Egress Interface.
 - (9) Now the Tester sends request to withdraw route A to DUT, TRx(Awith) also called WdrawTime1(Rt-A).
 - (10) Record the time when no traffic is observed on the Egress Interface. This is the RouteRemoveTime1(Rt-A).
 - (11) The difference between the RouteRemoveTime1 and WdrawTime1 is the WdrawConvTime1
- $$\text{WdrawConvTime1(Rt-A)} = \text{RouteRemoveTime1(Rt-A)} - \text{WdrawTime1(Rt-A)}$$

- C. Step 2:

- (1) Continuing from Step 1, re-advertise route A back to DUT from Tester.
- (2) The DUT will try to advertise the route A to Helper Node (This assumes 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 (e.g. route A). Traffic would be observed on the Egress interface after route A is received by the Helper Node

WATime=time traffic first flows

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

- (7) Total withdraw processing time is

TotalWdrawTime(Rt-A) = ((RouteRemoveTime2(Rt-A) - WdrawTime2(Rt-A)) - WdrawConvTime1(Rt-A))

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, route A 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 (e.g. route A). Initially traffic would be observed on the Best Egress interface.
- H. Now advertise the same route route A 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

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

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

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

$\text{Convergence} = \text{Path-switch time}(\text{Rt-A}) - \text{Path Change Event Time}(\text{Rt-A})$

- 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 System.s
- 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 route A 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 (e.g. route A). Initially no traffic would be observed on the Egress interface as the route A 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 route A is received on the Egress Interface

TRr(DUT, routeA). This is also called RecTime(Rt-A)

- R. The following equation represents the Graceful Restart Convergence Time

$\text{Graceful Restart Convergence Time(Rt-A)} = ((\text{RecTime(Rt-A)} - \text{GReventTime}) - \text{RIB-IN})$

- 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. Acknowledgements

We would like to thank Anil Tandon, Arvind Pandey, Mohan Nanduri, Jay Karthik and Eric Brendel, for their input and discussions on various sections in the document.

10. References

10.1. 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.

10.2. 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
Adara Networks

Email: shares@ndzh.com

Dean Lee
IXIA

Email: dlee@ixiacom.com

Ilya Varlashkin
Easynet Global Services

Email: ilya.varlashkin@easynet.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 5, 2013

M. Hamilton
Ixia
S. Banks
Aerohive Networks
Feb 2013

Benchmarking Methodology for Content-Aware Network Devices
draft-ietf-bmwg-ca-bench-meth-04

Abstract

This document defines a set of test scenarios and metrics that can be used to benchmark content-aware network devices. The scenarios in the following document are intended to more accurately predict the performance of these devices when subjected to dynamic traffic patterns. This document will operate within the constraints of the Benchmarking Working Group charter, namely black box characterization in a laboratory environment.

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 August 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
1.1. Requirements Language	5
2. Scope	5
3. Test Setup	5
3.1. Test Considerations	6
3.2. Clients and Servers	6
3.3. Traffic Generation Requirements	6
3.4. Discussion of Network Limitations	6
3.5. Framework for Traffic Specification	8
3.6. Multiple Client/Server Testing	8
3.7. Device Configuration Considerations	8
3.7.1. Network Addressing	9
3.7.2. Network Address Translation	9
3.7.3. TCP Stack Considerations	9
3.7.4. Other Considerations	9
4. Benchmarking Tests	9
4.1. Maximum Application Session Establishment Rate	10
4.1.1. Objective	10
4.1.2. Setup Parameters	10
4.1.3. Procedure	10
4.1.4. Measurement	10
4.1.4.1. Maximum Application Flow Rate	10
4.1.4.2. Application Flow Duration	11
4.1.4.3. Application Efficiency	11
4.1.4.4. Application Flow Latency	11
4.2. Application Throughput	11
4.2.1. Objective	11
4.2.2. Setup Parameters	11
4.2.3. Procedure	12
4.2.4. Measurement	12
4.2.4.1. Maximum Throughput	12
4.2.4.2. Maximum Application Flow Rate	12
4.2.4.3. Application Flow Duration	12
4.2.4.4. Application Efficiency	12
4.2.4.5. Packet Loss	12
4.2.4.6. Application Flow Latency	12
4.3. Malformed Traffic Handling	13
4.3.1. Objective	13
4.3.2. Setup Parameters	13
4.3.3. Procedure	13
4.3.4. Measurement	13

5. IANA Considerations	13
6. Security Considerations	13
7. References	14
7.1. Normative References	14
7.2. Informative References	15
7.3. URL References	15
Appendix A. Example Traffic Mix	15
Appendix B. Malformed Traffic Algorithm	17
Authors' Addresses	19

1. Introduction

Content-aware and deep packet inspection (DPI) device deployments have grown significantly in recent years. No longer are devices simply using Ethernet and IP headers to make forwarding decisions. This class of device now uses application-specific data to make these decisions. For example, a web-application firewall (WAF) may use search criteria upon the HTTP uniform resource indicator (URI)[1] to decide whether a HTTP GET method may traverse the network. In the case of lawful/legal intercept technology, a device could use the phone number within the Session Description Protocol[14] to determine whether a voice-over-IP phone may be allowed to connect. In addition to the development of entirely new classes of devices, devices that could historically be classified as 'stateless' or raw forwarding devices are now performing DPI functionality. Devices such as core and edge routers are now being developed with DPI functionality to make more intelligent routing and forwarding decisions.

The Benchmarking Working Group (BMWG) has historically produced Internet Drafts and Requests for Comment that are focused specifically on creating output metrics that are derived from a very specific and well-defined set of input parameters that are completely and unequivocally reproducible from test bed to test bed. The end goal of such methodologies is to, in the words of the RFC 2544 [2], reduce "specsmanship" in the industry and hold vendors accountable for performance claims.

The end goal of this methodology is to generate performance metrics in a lab environment that will closely relate to actual observed performance on production networks. By utilizing dynamic traffic patterns relevant to modern networks, this methodology should be able to closely tie laboratory and production metrics. It should be further noted that any metrics acquired from production networks SHOULD be captured according to the policies and procedures of the IPPM or PMOL working groups.

An explicit non-goal of this document is to replace existing methodology/terminology pairs such as RFC 2544 [2]/RFC 1242 [3] or RFC 3511 [4]/RFC 2647 [5]. The explicit goal of this document is to create a methodology more suited for modern devices while complementing the data acquired using existing BMWG methodologies. This document does not assume completely repeatable input stimulus. The nature of application-driven networks is such that a single dropped packet inherently changes the input stimulus from a network perspective. While application flows will be specified in great detail, it simply is not practical to require totally repeatable input stimulus.

1.1. 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 [6].

2. Scope

Content-aware devices take many forms, shapes and architectures. These devices are advanced network interconnect devices that inspect deep into the application payload of network data packets to do classification. They may be as simple as a firewall that uses application data inspection for rule set enforcement, or they may have advanced functionality such as performing protocol decoding and validation, anti-virus, anti-spam and even application exploit filtering. The document will universally call these devices middleboxes, as defined by RFC 3234 [7].

This document is strictly focused on examining performance and robustness across a focused set of metrics: throughput(min/max/avg/sample std dev), transaction rates(successful/failed), application response times, concurrent flows, and unidirectional packet latency. None of the metrics captured through this methodology are specific to a device and the results are DUT implementation independent. Functional testing of the DUT is outside the scope of this methodology.

Devices such as firewalls, intrusion detection and prevention devices, wireless LAN controllers, application delivery controllers, deep packet inspection devices, wide-area network(WAN) optimization devices, and unified threat management systems generally fall into the content-aware category. While this list may become obsolete, these are a subset of devices that fall under this scope of testing.

3. Test Setup

This document will be applicable to most test configurations and will not be confined to a discussion on specific test configurations. Since each DUT/SUT will have their own unique configuration, users SHOULD configure their device with the same parameters that would be used in the actual deployment of the device or a typical deployment, if the actual deployment is unknown. A summary of the DUT configuration MUST be published with the final benchmarking results. In order to improve repeatability, the published configuration information SHOULD include command-line scripts used to configure the DUT, if any, and SHOULD also include any configuration information

for the test equipment used."

3.1. Test Considerations

3.2. Clients and Servers

Content-aware device testing SHOULD involve multiple clients and multiple servers. As with RFC 3511 [4], this methodology will use the terms virtual clients/servers because both the client and server will be represented by the tester and not actual clients/servers. Similarly defined in RFC 3511 [4], a data source may emulate multiple clients and/or servers within the context of the same test scenario. The test report SHOULD indicate the number of virtual clients/servers used during the test. IANA has reserved address ranges for laboratory characterization. These are defined for IPv4 and IPv6 by RFC 2544 Appendix C [2] and RFC 5180 Section 5.2 [8] respectively and SHOULD be consulted prior to testing.

3.3. Traffic Generation Requirements

The explicit purposes of content-aware devices vary widely, but these devices use information deeper inside the application flow to make decisions and classify traffic. This methodology will utilize traffic flows that resemble real application traffic without utilizing captures from live production networks. Application Flows, as defined in Section 1.1 RFC 2724 [9] are able to be well-defined without simply referring to a network capture. An example traffic template is defined and listed in Appendix A of this document. A user of this methodology is free to utilize the example mix as provided in the appendix. If a user of this methodology understands the traffic patterns in their production network, that user MAY use the template provided in Appendix A to describe a traffic mix appropriate for their environment. In all cases, users MUST report the traffic mix used in the test, and SHOULD report this using a template similar to that in Appendix A.

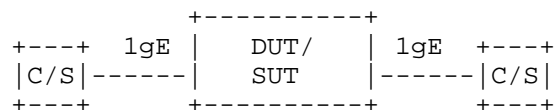
The test tool SHOULD be able to create application flows between every client and server, regardless of direction. The tester SHOULD be able to open TCP connections on multiple destination ports and SHOULD be able to direct UDP traffic to multiple destination ports.

3.4. Discussion of Network Limitations

Prior to executing the methodology as outlined in the following sections, it is imperative to understand the implications of utilizing representative application flows for the traffic content of the benchmarking effort. One interesting aspect of utilizing application flows is that each flow is inherently different from

every other application flow. The content of each flow will vary from application to application, and in most cases, even varies within the same type of application flow. The following description of the methodology will individually benchmark every individual type and subset of application flow, prior to performing similar tests with a traffic mix as specified either by the example mix in Appendix A, or as defined by the user of this methodology.

The purpose of this process is to ensure that any performance implications that are discovered during the mixed testing aren't due to the inherent physical network limitations. As an example of this phenomena, it is useful to examine a network device inserted into a single path, as illustrated in the following diagram.



Simple Inline DUT Configuration

Figure 1: Simple Middle-box Example

For the purpose of this discussion, let's take a hypothetical application flow that utilizes UDP for the transport layer. Assume that the sample transaction we will be using to model this particular flow requires 10 UDP datagrams to complete the transaction. For simplicity, each datagram within the flow is exactly 64 bytes, including associated Ethernet, IP, and UDP overhead. With any network device, there are always three metrics which interact with each other: number of concurrent application flows, number of application flows per second, and layer-7 throughput.

Our example test bed is a single-path device connected by 1 gigabit Ethernet links. The purpose of this benchmark effort is to quantify the number of application flows per second that may be processed through our device under test. Let's assume that the result from our scenario is that the DUT is able to process 10,000 application flows per second. The question is whether that ceiling is the actual ceiling of the device, or if it is actually being limited by one of the other metrics. If we do the appropriate math, 10000 flows per second, with each flow at 640 total bytes means that we are achieving an aggregate bitrate of roughly 49 Mbps. This is dramatically less than the 1 gigabit physical link we are using. We can conclude that 10,000 flows per second is in fact the performance limit of the device.

If we change the example slightly and increase the size of each

datagram to 1312 bytes, then it becomes necessary to recompute the load. Assuming the same observed DUT limitation of 10,000 flows per second, it must be ensured that this is an artifact of the DUT, and not of physical limitations. For each flow, we'll require 104,960 bits. 10,000 flows per second implies a throughput of roughly 1 Gbps. At this point, we cannot definitively answer whether the DUT is actually limited to 10,000 flows per second. If we are able to modify the scenario, and utilize 10 Gigabit interfaces, then perhaps the flow per second ceiling will be reached at a higher number than 10,000.

This example illustrates why a user of this methodology SHOULD benchmark each application variant individually to ensure that the cause of a measured limit is fully understood

3.5. Framework for Traffic Specification

The following table SHOULD be specified for each application flow variant.

- o Data Exchanged By Flow, Bits
- o Offered Percentage of Total Flows
- o Transport Protocol(s)
- o Destination Port(s)

3.6. Multiple Client/Server Testing

In actual network deployments, connections are being established between multiple clients and multiple servers simultaneously. Device vendors have been known to optimize the operation of their devices for easily defined patterns. The connection sequence ordering scenarios a device will see on a network will likely be much less deterministic. In fact, many application flows have multiple layer 4 connections within a single flow, with client and server reversing roles. Flow initiation SHOULD be in a pseudo-random manner across ingress ports.

3.7. Device Configuration Considerations

The configuration of the DUT may have an effect on the observed results of the following methodology. A comprehensive, but certainly not exhaustive, list of potential considerations is listed below.

3.7.1. Network Addressing

The IANA has issued a range of IP addresses to the BMWG for purposes of benchmarking. Please refer to RFC 2544 [2] and RFC 5180 [8] for more details. If more IPv4 addresses are required than the RFC 2544 allotment provides, then allocations from the private address space as defined in RFC 1918 [10] may be used.

3.7.2. Network Address Translation

Many content-aware devices are capable of performing Network Address Translation (NAT)[5]. If the final deployment of the DUT will have this functionality enabled, then the DUT SHOULD also have it enabled during the execution of this methodology. It MAY be beneficial to perform the test series in both modes in order to determine the performance differential when using NAT. The test report SHOULD indicate whether NAT was enabled during the testing process.

3.7.3. TCP Stack Considerations

The IETF has historically provided guidance and information on TCP stack considerations. This methodology is strictly focused on performance metrics at layers above 4, thus does not specifically define any TCP stack configuration parameters of either the tester or the DUTs. The TCP configuration of the tester MUST remain constant across all DUTs in order to ensure comparable results. While the following list of references is not exhaustive, each document contains a relevant discussion on TCP stack considerations.

The general IETF TCP roadmap is defined in RFC 4614 [11] and congestion control algorithms are discussed in Section 2 of RFC 3148 [12] with even more detailed references. TCP receive and congestion window sizes are discussed in detail in RFC 6349 [13].

3.7.4. Other Considerations

Various content-aware devices will have widely varying feature sets. In the interest of representative test results, the DUT features that will likely be enabled in the final deployment SHOULD be used. This methodology is not intended to advise on which features should be enabled, but to suggest using actual deployment configurations.

4. Benchmarking Tests

Each of the following benchmark scenarios SHOULD be run with each of the single application flow templates. Upon completion of all iterations, the mixed test SHOULD be completed, subject to the

traffic mix as defined by the user.

4.1. Maximum Application Session Establishment Rate

4.1.1. Objective

To determine the maximum rate through which a device is able to establish and complete application flows as defined by draft-ietf-bmwg-ca-bench-term-00.

4.1.2. Setup Parameters

The following parameters SHOULD be used and reported for all tests:

For each application protocol in use during the test run, the table provided in Section 3.5 SHOULD be published.

4.1.3. Procedure

The test SHOULD generate application network traffic that meets the conditions of Section 3.3. The traffic pattern SHOULD begin with an application flow rate of 10% of expected maximum. The test SHOULD be configured to increase the attempt rate in units of 10% up through 110% of expected maximum. In the case where expected maximum is limited by physical link rate as discovered through Appendix A, the maximum rate will attempted will be 100% of expected maximum, or "wire-speed performance". The duration of each loading phase SHOULD be at least 30 seconds. This test MAY be repeated, each subsequent iteration beginning at 5% of expected maximum and increasing session establishment rate to 110% of the maximum observed from the previous test run.

This procedure MAY be repeated any reasonable number of times with the results being averaged together.

4.1.4. Measurement

The following metrics MAY be determined from this test, and SHOULD be observed for each application protocol within the traffic mix:

4.1.4.1. Maximum Application Flow Rate

The test tool SHOULD report the maximum rate at which application flows were completed, as defined by RFC 2647 [5], Section 3.7. This rate SHOULD be reported individually for each application protocol present within the traffic mix.

4.1.4.2. Application Flow Duration

The test tool SHOULD report the minimum, maximum and average application duration, as defined by RFC 2647 [5], Section 3.9. This duration SHOULD be reported individually for each application protocol present within the traffic mix.

4.1.4.3. Application Efficiency

The test tool SHOULD report the application efficiency, similarly defined for TCP by RFC 6349 [13].

$$\text{App Efficiency \%} = \frac{\text{Transmitted Bytes} - \text{Retransmitted Bytes}}{\text{Transmitted Bytes}} \times 100$$

Figure 2: Application Efficiency Percent Calculation

Note that a calculation less than 100% does not necessarily imply noticeably degraded performance since certain applications utilize algorithms to maintain a quality user experience in the face of data loss.

4.1.4.4. Application Flow Latency

The test tool SHOULD report the minimum, maximum and average amount of time an application flow member takes to traverse the DUT, as defined by RFC 1242 [3], Section 3.8. This value SHOULD be reported individually for each application protocol present within the traffic mix.

4.2. Application Throughput

4.2.1. Objective

To determine the maximum rate through which a device is able to forward bits when using application flows as defined in the previous sections.

4.2.2. Setup Parameters

The same parameter reporting procedure as described in Section 4.1.2 SHOULD be used for all tests.

4.2.3. Procedure

This test will attempt to send application flows through the device at a flow rate of 30% of the maximum, as observed in Section 4.1. This procedure MAY be repeated with the results from each iteration averaged together.

4.2.4. Measurement

The following metrics MAY be determined from this test, and SHOULD be observed for each application protocol within the traffic mix:

4.2.4.1. Maximum Throughput

The test tool SHOULD report the minimum, maximum and average application throughput.

4.2.4.2. Maximum Application Flow Rate

The test tool SHOULD report the maximum rate at which application flows were completed, as defined by RFC 2647 [5], Section 3.7. This rate SHOULD be reported individually for each application protocol present within the traffic mix.

4.2.4.3. Application Flow Duration

The test tool SHOULD report the minimum, maximum and average application duration, as defined by RFC 2647 [5], Section 3.9. This duration SHOULD be reported individually for each application protocol present within the traffic mix.

4.2.4.4. Application Efficiency

The test tool SHOULD report the application efficiency as defined in Section 4.1.4.3.

4.2.4.5. Packet Loss

The test tool SHOULD report the number of packets lost or dropped from source to destination.

4.2.4.6. Application Flow Latency

The test tool SHOULD report the minimum, maximum and average amount of time an application flow member takes to traverse the DUT, as defined by RFC 1242 [3], Section 3.13. This value SHOULD be reported individually for each application protocol present within the traffic mix.

4.3. Malformed Traffic Handling

4.3.1. Objective

To determine the effects on performance and stability that malformed traffic may have on the DUT.

4.3.2. Setup Parameters

The same parameters SHOULD be used for Transport-Layer and Application Layer Parameters previously specified in Section 4.1.2 and Section 4.2.2.

4.3.3. Procedure

This test will utilize the procedures specified previously in Section 4.1.3 and Section 4.2.3. When performing the procedures listed previously, the tester should generate malformed traffic at all protocol layers. This is commonly known as fuzzed traffic. Fuzzing techniques generally modify portions of packets, including checksum errors, invalid protocol options, and improper protocol conformance.

The process by which the tester SHOULD generate the malformed traffic is outlined in detail in Appendix B.

4.3.4. Measurement

For each protocol present in the traffic mix, the metrics specified by Section 4.1.4 and Section 4.2.4 MAY be determined. This data may be used to ascertain the effects of fuzzed traffic on the DUT.

5. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see the update of RFC 2434 [15] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

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 RFC 2544 [2].

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 mis-route traffic to the test management network

7. References

7.1. Normative References

- [1] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [2] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.
- [3] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [4] Hickman, B., Newman, D., Tadjudin, S., and T. Martin, "Benchmarking Methodology for Firewall Performance", RFC 3511, April 2003.
- [5] Newman, D., "Benchmarking Terminology for Firewall Performance", RFC 2647, August 1999.
- [6] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [7] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.
- [8] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, May 2008.
- [9] Handelman, S., Stibler, S., Brownlee, N., and G. Ruth, "RTFM: New Attributes for Traffic Flow Measurement", RFC 2724, October 1999.
- [10] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

- [11] Duke, M., Braden, R., Eddy, W., and E. Blanton, "A Roadmap for Transmission Control Protocol (TCP) Specification Documents", RFC 4614, September 2006.
- [12] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.
- [13] Constantine, B., Forget, G., Geib, R., and R. Schrage, "Framework for TCP Throughput Testing", RFC 6349, August 2011.

7.2. Informative References

- [14] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [15] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

7.3. URL References

- [16] Sandvine Corporation, "<http://www.sandvine.com/general/document.download.asp?docID=58&sourceID=0>", 2012.

Appendix A. Example Traffic Mix

This appendix shows an example case of a protocol mix that may be used with this methodology. This mix closely represents the research published by Sandvine [16] in their biannual report for the first half of 2012 on North American fixed access service provider networks.

Direction	Application Flow	Options	Value
Upstream	BitTorrent	Avg Flow Size (L7)	512 MB
		Flow Percentage	44.4%
	HTTP	Avg Flow Size (L7)	128 kB
		Flow Percentage	7.3%
	Skype	Avg Flow Size (L7)	8 MB
		Flow Percentage	4.9%
	SSL/TLS	Avg Flow Size (L7)	128 kB
		Flow Percentage	3.2%
	Netflix		

Downstream		PPStream	Avg Flow Size (L7)	500 kB
			Flow Percentage	3.1%
		YouTube	Avg Flow Size (L7)	500 MB
			Flow Percentage	2.2%
		Facebook	Avg Flow Size (L7)	4 MB
			Flow Percentage	1.9%
		Teredo	Avg Flow Size (L7)	2 MB
			Flow Percentage	1.9%
		Apple iMessage	Avg Flow Size (L7)	500 MB
			Flow Percentage	1.2%
		Bulk TCP	Avg Flow Size (L7)	40 kB
			Flow Percentage	1.1%
		Netflix	Avg Flow Size (L7)	128 kB
			Flow Percentage	28.8%
		YouTube	Avg Flow Size (L7)	512 MB
			Flow Percentage	32.9%
		HTTP	Avg Flow Size (L7)	5 MB
			Flow Percentage	13.8%
		BitTorrent	Avg Flow Size (L7)	1 MB
			Flow Percentage	12.1%
		iTunes	Avg Flow Size (L7)	500 MB
			Flow Percentage	6.3%
		Flash Video	Avg Flow Size (L7)	32 MB
			Flow Percentage	3.8%
		MPEG	Avg Flow Size (L7)	100 MB
			Flow Percentage	2.6%
		RTMP	Avg Flow Size (L7)	100 MB
			Flow Percentage	2.0%
		Hulu	Avg Flow Size (L7)	50 MB
			Flow Percentage	2.0%
		SSL/TLS	Avg Flow Size (L7)	300 MB
			Flow Percentage	1.8%

	Bulk TCP	Avg Flow Size (L7)	256 kB
		Flow Percentage	1.6%
		Avg Flow Size (L7)	500 kB
		Flow Percentage	21.1%

Table 1: Example Traffic Pattern

Appendix B. Malformed Traffic Algorithm

Each application flow will be broken into multiple transport segments, IP packets, and Ethernet frames. The malformed traffic algorithm looks very similar to the IP Stack Integrity Checker project at <http://isic.sourceforge.net>.

The algorithm is very simple and starts by defining each of the fields within the TCP/IP stack that will be malformed during transmission. The following table illustrates the Ethernet, IPv4, IPv6, TCP, and UDP fields which are able to be malformed by the algorithm. The first column lists the protocol, the second column shows the actual header field name, with the third column showing the percentage of packets that should have the field modified by the malformation algorithm.

Protocol	Header Field	Malformed %
Total Frames		1%
Ethernet	Destination MAC	0%
	Source MAC	1%
	Ethertype	1%
	CRC	1%
IP Version 4	Version	1%
	IHL	1%
	Type of Service	1%
	Total Length	1%
	Identification	1%
	Flags	1%
	Fragment Offset	1%
	Time to Live	1%
	Protocol	1%
	Header Checksum	1%
	Source Address	1%
	Destination Address	1%
	Options	1%
	Padding	1%
UDP	Source Port	1%
	Destination Port	1%
	Length	1%
	Checksum	1%
TCP	Source Port	1%
	Destination Port	1%
	Sequence Number	1%
	Acknowledgement Number	1%
	Data Offset	1%
	Reserved(3 bit)	1%
	Flags(9 bit)	1%
	Window Size	1%
	Checksum	1%
	Urgent Pointer	1%
	Options(Variable Length)	1%

Table 2: Malformed Header Values

This algorithm is to be used across the regular application flows used throughout the rest of the methodology. As each frame is emitted from the test tool, a pseudo-random number generator will

indicate whether the frame is to be malformed by creating a number between 0 and 100. If the number is less than the percentage defined in the table, then that frame will be malformed. If the frame is to be malformed, then each of the headers in the table present within the frame will follow the same process. If it is determined that a header field should be malformed, the same pseudo-random number generator will be used to create a random number for the specified header field.

Authors' Addresses

Mike Hamilton
Ixia
Austin, TX 78730
US

Phone: +1 512 636 2303
Email: mhamilton@ixiacom.com

Sarah Banks
Aerohive Networks
San Jose, CA 95134
US

Email: sbanks@aerohive.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	dcba

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/>

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.

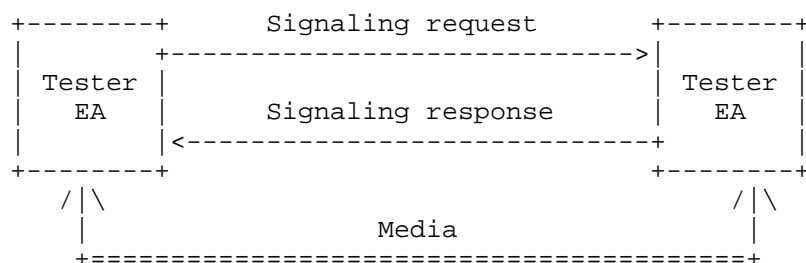


Figure 1: Baseline performance of the Emulated Agent without a DUT present

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.

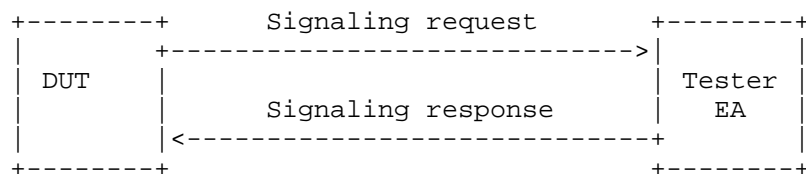


Figure 2: Baseline performance for DUT acting as a user agent client without associated media

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.

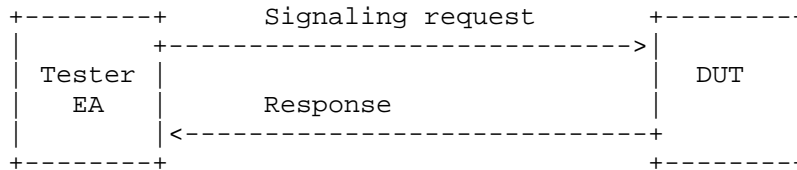


Figure 3: Baseline performance for DUT acting as a user agent server without associated media

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.

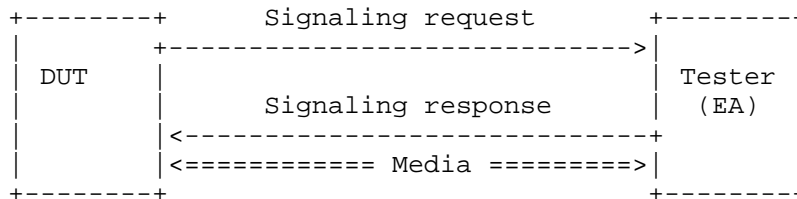


Figure 4: Baseline performance for DUT acting as a user agent client with associated media

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.

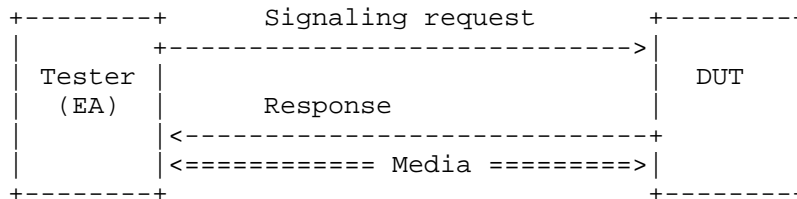


Figure 5: Baseline performance for DUT acting as a user agent server

with associated media

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

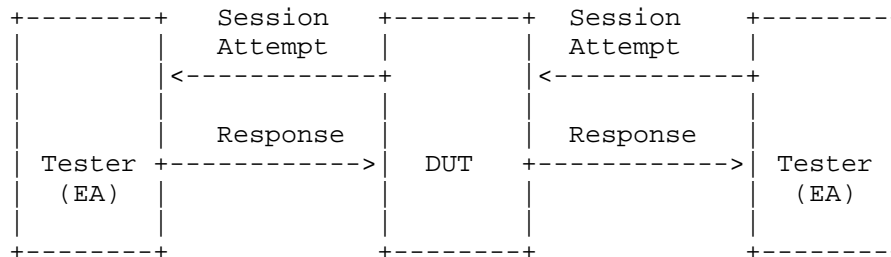


Figure 6: DUT/SUT performance benchmark for session establishment without media

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

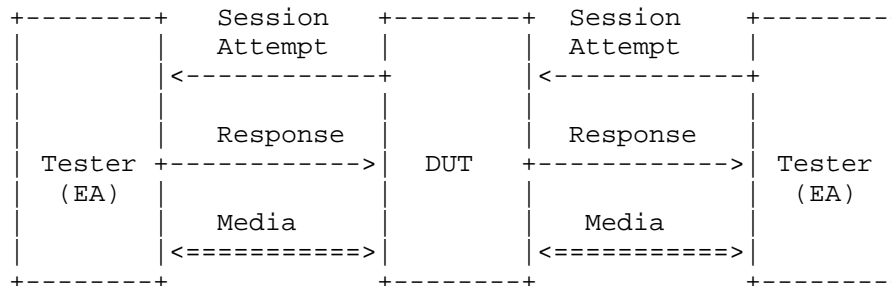


Figure 7: DUT/SUT performance benchmark for session establishment with media traversing the DUT

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.

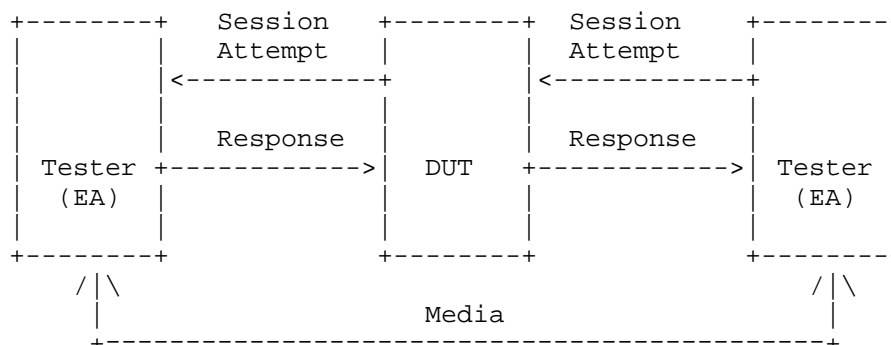


Figure 8: DUT/SUT performance benchmark for session establishment with media external to the DUT

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.

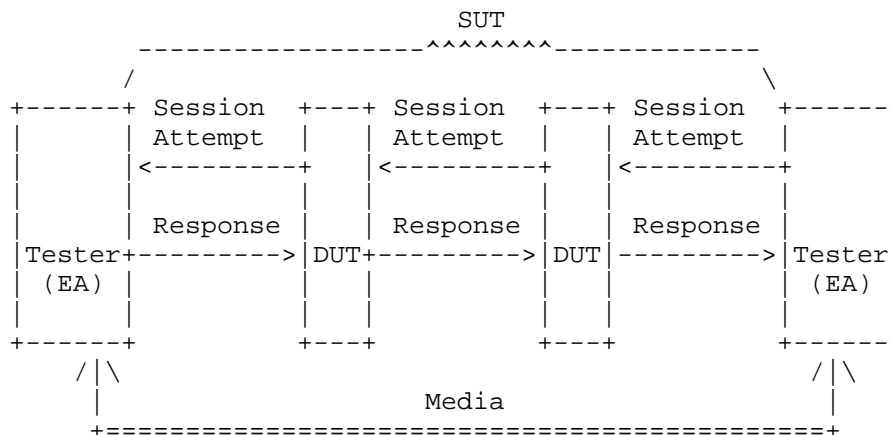


Figure 9: DUT/SUT performance benchmark for session establishment with multiple DUTs and end-to-end media

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.

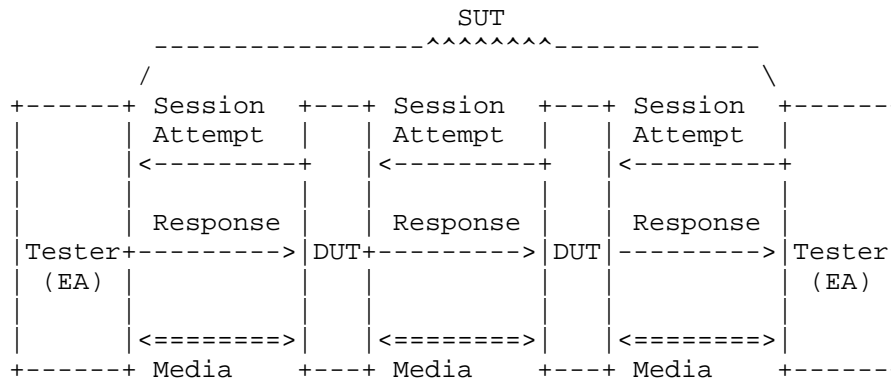


Figure 10: DUT/SUT performance benchmark for session establishment with multiple DUTs and hop-by-hop media

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.)

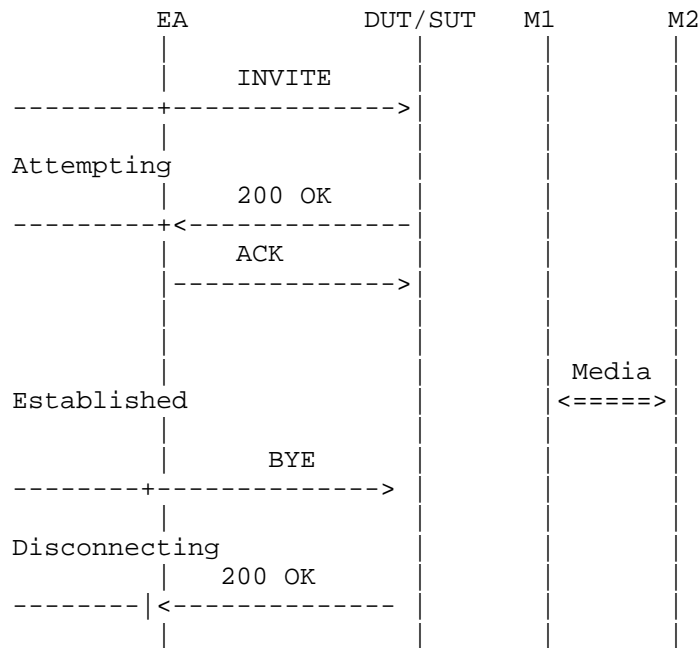


Figure 11: Invite-initiated Session States

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)

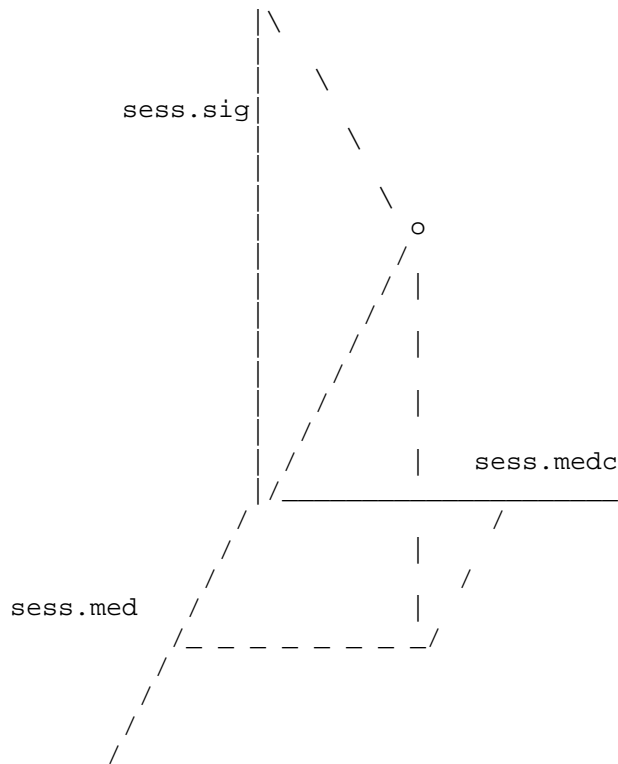


Figure 12: Session components

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: January 4, 2014

B. Parise
Cisco Systems
R. Papneja
Huawei Technologies
July 3, 2013

Terminology for Benchmarking LDP Data Plane Convergence
draft-parise-bmwg-ldp-convergence-term-00.txt

Abstract

This document defines new terms for benchmarking of LDP convergence. These terms are to be used in future methodology documents for benchmarking LDP Convergence. Existing BMWG terminology documents such as IGP Convergence Benchmarking [RFC 6412] provide useful terms for LDP Convergence benchmarking. These terms are discussed in this document. Applicable terminology for MPLS and LDP defined in MPLS WG RFCs [RFC 3031] and [RFC 5036] are also discussed.

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 January 4, 2014.

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 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
2. Existing Definitions	4
2.1. BMWG Convergence Terms	4
2.2. MPLS/LDP Terms	4
3. Term Definitions	5
3.1. LDP Binding Table	5
3.2. FEC Forwarding Table	6
3.3. FEC Convergence Event	6
3.4. FEC Forwarding Table Convergence	7
3.5. FEC Convergence	7
3.6. Multiple Next-Hop FEC	8
3.7. Ingress LSR	9
3.8. Egress LSR	9
3.9. LDP Peer	10
3.10. Targeted LDP Peer	11
3.11. Targeted FECs	11
3.12. Multi-Labeled Packets	12
3.13. Equal Cost Multiple Paths	12
3.14. Equal Cost Multiple FECs	13
3.15. FEC Convergence at Ingress LSR	13
3.16. FEC Convergence at Midpoint LSR	14
3.17. LDP Advertisement Type	14
3.18. Label Merging LSR	15
3.19. Non-merging LSR	16
3.20. LDPv6	16
4. Factors impacting Convergence	17
4.1. Interaction with Other Protocols	17
4.2. Timers	17
4.3. TCP Parameters	17
5. Security Considerations	17
6. Acknowledgements	17
7. References	18
7.1. Normative References	18
7.2. Informative References	18
Authors' Addresses	18

1. Introduction

This draft describes the terminology for benchmarking LDP Convergence. An accompanying document will describe the methodology for doing the benchmarking. The main motivation for doing this work is the increased focus on lowering convergence time for LDP as an alternative to other solutions such as MPLS Fast Reroute (i.e. protection techniques using RSVP-TE extensions).

The purpose of this document is to find existing terminology as well as define new terminology when needed terms are not available. The terminology will support the methodology that will be based on black-box testing of the LDP dataplane. The approach is very similar to the one found in [RFC 6412] and [RFC 6413].

2. Existing Definitions

2.1. BMWG Convergence Terms

This document uses existing terminology defined in other IETF documents. These include the following:

Route Convergence	Defined in [RFC 6412]
Convergence Packet Loss	Defined in [RFC 6412]
Convergence Event Instant	Defined in [RFC 6412]
Convergence Recovery Instant	Defined in [RFC 6412]
Rate-Derived Convergence Time	Defined in [RFC 6412]
Convergence Event Transition	Defined in [RFC 6412]
Convergence Recovery Transition	Defined in [RFC 6412]
Loss-Derived Convergence Time	Defined in [RFC 6412]
Restoration Convergence Time	Defined in [RFC 6412]
Packet Sampling Interval	Defined in [RFC 6412]
Local Interface	Defined in [RFC 6412]
Neighbor Interface	Defined in [RFC 6412]
Remote Interface	Defined in [RFC 6412]
Preferred Egress Interface	Defined in [RFC 6412]
Next-Best Egress Interface	Defined in [RFC 6412]
Stale Forwarding	Defined in [RFC 6412]

2.2. MPLS/LDP Terms

Label	Defined in [RFC 3031]
FEC	Defined in [RFC 3031]
Label Withdraw	Defined in [RFC 5036]
LSP	Defined in [RFC 3031]
LSR	Defined in [RFC 3031]
LDP Identifier	Defined in [RFC 5036]
LDP Session	Defined in [RFC 5036]
Per-Interface Label Space	Defined in [RFC 3031]
Per-Platform Label Space	Defined in [RFC 3031]
MPLS Node	Defined in [RFC 3031]
MPLS Edge Node	Defined in [RFC 3031]
MPLS Egress Node	Defined in [RFC 3031]
MPLS Ingress Node	Defined in [RFC 3031]
Upstream LSR	Defined in [RFC 3031]
Downstream LSR	Defined in [RFC 3031]
Local Repair	Defined in [RFC 4090]
PLR	Defined in [RFC 4090]
One-to-One Backup	Defined in [RFC 4090]
Detour LSP	Defined in [RFC 4090]
Backup Path	Defined in [RFC 4090]
Downstream-on-Demand	Defined in [RFC 3031]
Unsolicited Downstream	Defined in [RFC 3031]
Independent Label Distribution Control	Defined in [RFC 5036]
Address Family	Defined in [RFC 5036]
IGP Update Message	ISIS/OSPF LSA

3. Term Definitions

3.1. LDP Binding Table

Definition:

Table in which the LSR maintains all learned labels. It consists of the prefix and label information bound to a peer's LDP identifier and the list of sent and received bindings/peer.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

FEC Forwarding Table

3.2. FEC Forwarding Table

Definition:

Table in which the LSR maintains the next hop information for the particular FEC with the associated outgoing label and interface. The information used for setting up the FEC forwarding table is retrieved from the LDP Binding Table.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

LDP Binding Table

3.3. FEC Convergence Event

Definition:

The occurrence of a planned or unplanned action in the network that results in a change to an LSR's LDP next-hop forwarding.

Discussion:

Convergence Events include link loss, routing protocol session loss, router failure, and better next-hop. Also, different types of administrative events such as interface shutdown is considered.

Measurement Units:

N/A

Issues:

None

See Also:

FEC Forwarding Table Convergence

FEC Convergence

3.4. FEC Forwarding Table Convergence

Definition:

Recovery from a FEC Convergence Event that causes the FEC Forwarding Table to change and re-stabilize.

Discussion:

FEC Forwarding Table Convergence updates after the RIB and LDP Binding Table update due to a FEC Convergence Event. FEC Forwarding Table Convergence can be observed externally by the rerouting of data Traffic to a new egress interface.

Measurement Units:

seconds

Issues:

None

See Also:

FEC Forwarding Table

FEC Convergence Event

FEC Convergence

3.5. FEC Convergence

Definition:

Recovery from a FEC Convergence Event that causes the LDP Binding Table to change and re-stabilize.

Discussion:

FEC Convergence is a change in an LDP Binding of a prefix and label to a peer's LDP Identifier. This change can be an update or recovery due to a FEC Convergence Event. FEC Convergence is an LSR action made prior to FEC Forwarding Table Convergence. FEC Convergence is not an externally observable Black-Box measurement.

Measurement Units:

N/A

Issues:

Where is LDP Identifier defined? Where is LDP Binding defined?

See Also:

FEC Binding Table

FEC Convergence Event

FEC Forwarding Table Convergence

3.6. Multiple Next-Hop FEC

Definition:

A FEC with more than one next-hop and associated outgoing label and interface.

Discussion:

A Multiple Next-Hop FEC can be verified from the FEC Forwarding Table and from externally observing traffic being forwarded to a FEC on one or more interfaces.

Measurement Units:

N/A

Issues:

None

See Also:

FEC Forwarding Table

3.7. Ingress LSR

Definition:

An MPLS ingress node which is capable of forwarding native L3 packets.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

MPLS Node

MPLS Edge Node

MPLS Egress Node

MPLS Ingress Node

Label Switching Router (LSR)

Egress LSR

3.8. Egress LSR

Definition:

An MPLS Egress node which is capable of forwarding native L3 packets.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

MPLS Node

MPLS Edge Node

MPLS Egress Node

MPLS Ingress Node

Label Switching Router (LSR)

Ingress LSR

3.9. LDP Peer

Definition:

An adjacent LSR with which LDP adjacency is established

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

Targeted LDP Peer

3.10. Targeted LDP Peer

Definition:

An adjacent LSR (usually more than a hop away) with which LDP adjacency is established through a directed hello message which is unicast.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

LDP Peer

3.11. Targeted FECs

Definition:

The FECs advertised by a Targeted LDP Peer

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

Targeted Peer

3.12. Multi-Labeled Packets

Definition:

A data packet that has more than one label in the label stack.

Discussion:

This typically happens when a Targeted Peer is established over a traffic engineered tunnel.

Measurement Units:

N/A

Issues:

None

See Also:

None

3.13. Equal Cost Multiple Paths

Definition:

Existence of multiple IGP paths to reach a particular destination. In this case the depending on the implementation traffic destined to a prefix that has multiple equal cost paths is load balanced across all these paths.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

Equal Cost Multiple FECs

3.14. Equal Cost Multiple FECs

Definition:

Existence of multiple to reach a destination. Typically the LSR that has multiple FECs of equal costs does a load balance on all the FECs

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

Equal Cost Multiple Paths

3.15. FEC Convergence at Ingress LSR

Definition:

Recovery from a FEC Convergence Event that causes the LDP Binding Table to change and re-stabilize at the Ingress LSR

Discussion:

FEC Convergence is a change in an LDP Binding of a prefix and label to a peer's LDP Identifier. This change can be an update or recovery due to a FEC Convergence Event. FEC Convergence is an LSR action made prior to FEC Forwarding Table Convergence. FEC Convergence is not an externally observable Black-Box measurement.

Measurement Units:

N/A

Issues:

Where is LDP Identifier defined? Where is LDP Binding defined?

See Also:

LDP Binding Table

FEC Convergence Event

FEC Forwarding Table Convergence

3.16. FEC Convergence at Midpoint LSR

Definition:

Recovery from a FEC Convergence Event that causes the LDP Binding Table to change and re-stabilize at a Midpoint LSR

Discussion:

FEC Convergence is a change in an LDP Binding of a prefix and label to a peer's LDP Identifier. This change can be an update or recovery due to a FEC Convergence Event. FEC Convergence is an LSR action made prior to FEC Forwarding Table Convergence. FEC Convergence is not an externally observable Black-Box measurement.

Measurement Units:

N/A

Issues:

Where is LDP Identifier defined? Where is LDP Binding defined?

See Also:

LDP Binding Table

FEC Convergence Event

FEC Forwarding Table Convergence

3.17. LDP Advertisement Type

Definition:

The type of LDP advertisement in operation. Downstream On Demand vs Downstream Unsolicited.

Discussion:

None

Measurement Units:

N/A

Issues:

None

See Also:

None

3.18. Label Merging LSR

Definition:

A LSR which is capable of sending multiple packets out of the same outgoing interface with the same label even though it receives these packets from different incoming interfaces and may also receive them with the same lane

Discussion:

With label merging the LSR need to send a single label per FEC and also on the receiving end the number of incoming labels per FEC is never larger than the number of label distribution adjacencies

Measurement Units:

N/A

Issues:

There maybe be scenarios where a Merging LSR is capable of merging only a subset of incoming labels into a single outgoing label

See Also:

Non-Merging LSR and [RFC 3031]

3.19. Non-merging LSR

Definition:

A LSR which forwards packets with multiple outgoing labels when it receives packets from the same FEC with different incoming labels

Discussion:

Without label merging the number of outgoing labels per FEC could be as large as the number of nodes in the network

Measurement Units:

N/A

Issues:

None

See Also:

Label Merging LSR and [RFC 3031]

3.20. LDPv6

Definition:

This term implies forwarding of IPv6 packets as detailed in [RFC 5036]

Discussion:

None

Measurement Units:

N/A

Issues:

The current specification [RFC 5036] has certain gaps as detailed in [LDPv6]. Once its standardized we will extend the scope to cover those details.

See Also:

None

4. Factors impacting Convergence

4.1. Interaction with Other Protocols

LDP convergence must include the affect of interaction with IGPs. All test reports must include the IGPs provisioned in the test and their associated parameters

4.2. Timers

LDP convergence is impacted by the Hold and Keepalive Timers. Test reports must include all the relevant timer values

4.3. TCP Parameters

As LDP uses TCP for sessions, all relevant TCP session parameters must be reported

5. 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.

6. Acknowledgements

We thank Al Morton for providing valuable comments to this document. We also thank Scott Poretsky for his contributions to the initial version of this document.

7. References

7.1. Normative References

- [I-D.ietf-mpls-ldp-ipv6]
Asati, R., Manral, V., Papneja, R., and C. Pignataro,
"Updates to LDP for IPv6", draft-ietf-mpls-ldp-ipv6-08
(work in progress), February 2013.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol
Label Switching Architecture", RFC 3031, January 2001.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute
Extensions to RSVP-TE for LSP Tunnels", RFC 4090,
May 2005.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP
Specification", RFC 5036, October 2007.
- [RFC6412] Poretsky, S., Imhoff, B., and K. Michielsen, "Terminology
for Benchmarking Link-State IGP Data-Plane Route
Convergence", RFC 6412, November 2011.
- [RFC6413] Poretsky, S., Imhoff, B., and K. Michielsen, "Benchmarking
Methodology for Link-State IGP Data-Plane Route
Convergence", RFC 6413, November 2011.

7.2. Informative References

- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter,
"Multiprotocol Extensions for BGP-4", RFC 4760,
January 2007.

Authors' Addresses

Bhavani Parise
Cisco Systems

Email: bhavani@cisco.com

Rajiv Papneja
Huawei Technologies

Email: rajiv.papneja@huawei.com

