

Benchmarking Methodology WG  
Internet Draft  
Intended status: Informational  
Expires: March 3, 2014

Sarah Banks  
Aerohive Networks  
Fernando Calabria  
Cisco  
Gery Czirjak  
Ramdas Machat  
Juniper  
October 6,  
2013

ISSU Benchmarking Methodology  
draft-banks-bmwg-issu-meth-02

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
5.1	Pre-ISSU recommended verifications .....	9

5.2	Software Staging .....	11
5.3	Upgrade Run.....	12
5.4	Post ISSU verifications.....	12
5.5	ISSU under negative stimuli.....	12
6	ISSU Abort and Rollback .....	14
7	Final Report - Data Presentation - Analysis.....	14
8	Security Considerations.....	16
9	IANA Considerations.....	17
10	Conclusions.....	17
11	References.....	17
11.1	Normative References.....	17
11.2	Informative References.....	17
12	Acknowledgments.....	17

## 1. Introduction

As required by most Service Provider (SP) network operators, ISSU functionality has been implemented by modern forwarding devices to upgrade or downgrade from one software version to another with a goal of eliminating the downtime of the router and/or the outage of service. However, most operators expect that while elimination is the goal, minimal downtime and/or degradation of service is often expected. 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 or vice versa.
- 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 forwarding device'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 may consist of at least one pair of Routing Processors (RP's) that 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. The downloading of software process may be performed automatically by the device 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, a switchover of RPs may take place, where one RP is now upgraded with the new version of software. More importantly, the "Upgrade Run" phase is where the internal changes made to information and state stored on the router, on disk and in memory, are either migrated to the "new" version of code, or transformed/rebuilt to meet the standards of the new version of code, and pushed onto the appropriate pieces of hardware. It is within this phase that any outage(s) on the control or forwarding plane MAY be expected to be observed.

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

#### 5.5 ISSU under negative stimuli

As an OPTIONAL Test Case, the operator may want to perform an ISSU test while the DUT is under stress by introducing route churn to any or all of the involved phases of the ISSU process.

One approach relies on the operator to gather statistical information from the production environment and determine a specific number of routes to flap every 'fixed' or 'variable' interval. Alternatively, the operator may wish to simply pre-select a fixed number of prefixes to flap. As an example, an operator may decide to flap 1% of all the BGP routes every minute and restore them 1 minute afterwards. The tester may wish to apply this negative stimulus throughout the entire ISSU process or most importantly, during the run phase.

It is important to ensure that these routes, which are introduced solely for stress proposes, MUST not overlap the ones (per the Load Model) specifically leveraged to calculate the TP (recorded outage). Furthermore, there SHOULD NOT be 'operator induced' control plane - protocol adjacency flaps for the duration of the test process as it may adversely affect the characterization of

the entire test exercise. For example, triggering IGP adjacency events may force re-computation of underlying routing tables with attendant impact to the perceived ISSU timings. While not recommended, if such trigger events are desired by the test operator, care should be taken to avoid the introduction of unexpected anomalies within the test harness.

## 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.
Number of Prefixes flapped (ON Interval) (Optional)	# of prefixes / Time (minutes)
Number of Prefixes flapped (OFF Interval) (Optional)	# of prefixes / Time (minutes)

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](mailto:sbanks@aerohive.com)

Fernando Calabria  
Cisco Systems  
Email: [fcalabri@cisco.com](mailto:fcalabri@cisco.com)

Gery Czirjak  
Juniper Networks  
Email: [gczirjak@juniper.net](mailto:gczirjak@juniper.net)

Ramdas Machat  
Juniper Networks  
Email: [rmachat@juniper.net](mailto:rmachat@juniper.net)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 10, 2014

L. Avramov  
Cisco Systems  
J. Rapp  
Hewlett Packard  
November 6, 2013

Data Center Benchmarking Methodology  
draft-bmwg-dcbench-methodology-02

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

This Internet-Draft will expire on May 10, 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 . . . . .	7
3.1 Objective . . . . .	7
3.2 Methodology . . . . .	7
3.3 Reporting format . . . . .	10
4 Microburst Testing . . . . .	10
4.1 Objective . . . . .	10
4.2 Methodology . . . . .	10
4.3 Reporting Format . . . . .	11
5. Head of Line Blocking . . . . .	11
5.1 Objective . . . . .	11
5.2 Methodology . . . . .	11
5.3 Reporting Format . . . . .	13
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 SHOULD 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.

Alternatively when a traffic generator CAN NOT be connected to all ports on the DUT, a snake test MUST be used for line rate testing, excluding latency and jitter as those became then irrelevant. The snake test consists in the following method: -connect the first and last port of the DUT to a traffic generator-connect back to back sequentially all the ports in between: port 2 to 3, port 4 to 5 etc to port n-2 to port n-1; where n is the total number of ports of the DUT-configure port 1 and 2 in the same vlan X, port 3 and 4 in the same vlan Y, etc. port n-1 and port n in the same vlan ZZZ. This snake test provides a capability to test line rate for Layer 2 and Layer 3 RFC 2544/3918 in instance where a traffic generator with only two ports is available. The latency and jitter are not to be considered with this 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,1518b,4096,8000,9216b.

The pattern for testing can be expressed using RFC 6985 [IMIX Genome: Specification of Variable Packet Sizes for Additional Testing]

- throughput needs to be expressed in % of total transmitted frames

- 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 typical|many|multiple 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

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

Jacob Rapp  
Hewlett-Packard  
3000 Hanover Street  
Palo Alto, CA  
United States  
Phone: +1 650 857 3367  
Email: jacob.h.rapp@hp.com

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

W. Cervený  
Arbor Networks  
October 17, 2013

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

Abstract

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

Requirements Language

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

Status of This Memo

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

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

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

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

Copyright Notice

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. Overview of Relevant NDP and Intermediate Node Behavior . . . .	3
4. Test Setup . . . . .	5
4.1. Testing Interfaces . . . . .	6
5. Modifiers (variables) . . . . .	6
5.1. Frequency of NDP triggering packets . . . . .	6
6. Tests . . . . .	7
6.1. Maximum number of valid hosts . . . . .	7
6.1.1. Test Streams . . . . .	7
6.1.2. General Testing Procedure . . . . .	7
6.1.3. Discussion . . . . .	8
6.2. Stable-state time . . . . .	8
6.2.1. Test streams . . . . .	8
6.2.2. General Testing Procedure . . . . .	8
6.2.3. Discussion . . . . .	9
6.3. NDP Prioritization: Behavior with stale neighbor entries .	9
6.3.1. Test Streams . . . . .	9
6.3.2. General Testing Procedure . . . . .	9
6.3.3. Discussion . . . . .	10
6.4. NDP Prioritization: Entries never present in neighbor cache . . . . .	10
6.4.1. Test Streams . . . . .	10
6.4.2. General Testing Procedure . . . . .	10
6.4.3. Discussion . . . . .	10
6.5. NDP Prioritization: Unreachable addresses only . . . . .	10
6.5.1. Test Streams . . . . .	10
6.5.2. General Testing Procedure . . . . .	10
6.5.3. Discussion . . . . .	10
7. Measurements Explicitly Excluded . . . . .	11
7.1. DUT CPU Utilization . . . . .	11
7.2. Malformed Packets . . . . .	11
8. DUT initialization . . . . .	11
9. IANA Considerations . . . . .	11
10. Security Considerations . . . . .	11
11. Acknowledgements . . . . .	12
12. Normative References . . . . .	12
Author's Address . . . . .	12

## 1. Introduction

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

## 2. Terminology

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

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

**Neighbor Discovery** See Section 1 of RFC 4861 [RFC4861]

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

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

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

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

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

## 3. Overview of Relevant NDP and Intermediate Node Behavior

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

1. The intermediate node first determines if the destination IPv6 address is present in its neighbor cache.
2. If the address is present in the neighbor cache, the intermediate node forwards the packet to the destination node using the appropriate link-layer address and interface.
3. If the destination IPv6 address is not in the intermediate node's neighbor cache:
  1. An entry for the IPv6 address is added to the neighbor cache and the entry is marked "INCOMPLETE".
  2. The intermediate node sends a neighbor solicitation packet to the solicited-node multicast address on the interface considered on-link.
  3. If a solicited neighbor advertisement for the IPv6 address is received by the intermediate node, the neighbor cache entry is marked "REACHABLE" and remains in this state for 30 seconds.
  4. If a neighbor advertisement is not received, the intermediate node will continue sending neighbor solicitation packets every second until either a neighbor solicitation is received or the maximum number of solicitations has been sent. If a neighbor advertisement is not received in this period, the entry can be discarded.

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

1. There are a large number of real nodes connected via an intermediate node's interface and a large number of these nodes are sending and receiving traffic simultaneously.
2. There are a large number of addresses for which a scanning activity is occurring and no real node will respond to the neighbor solicitation. This scanning activity can be unintentional or malicious. In addition to maintaining the "INCOMPLETE" neighbor cache entry, the intermediate node must send a neighbor solicitation packet every second for the maximum number of solicitations. With today's network link bandwidths, a scanning event could cause a lot of entries to be added to the neighbor cache and solicited for in the time that it takes for a neighbor cache entry to be discarded.

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

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

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

#### 4. Test Setup

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

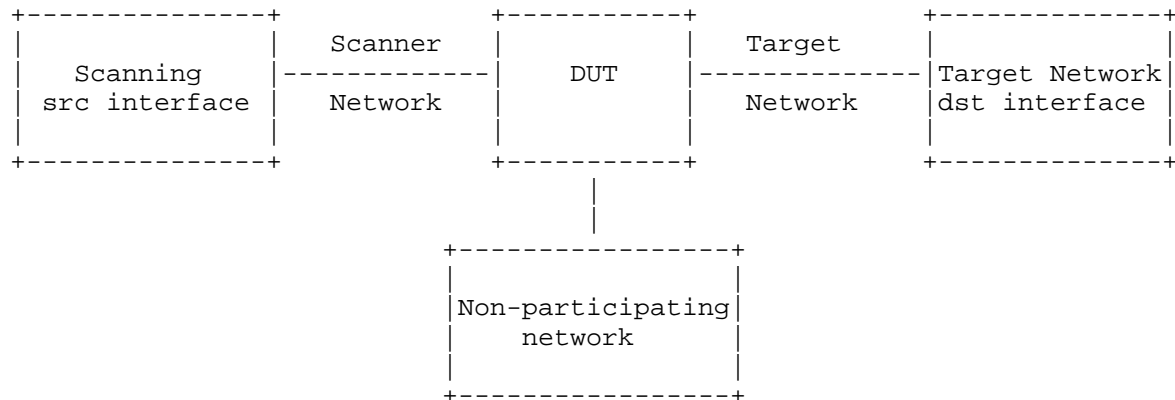
It is assumed that the latency for all network segments is negligible. By default, the target network's subnet shall be 64-bits in length, although some tests may involve increasing the prefix length.

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

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

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

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



#### 4.1. Testing Interfaces

Two tester interfaces are configured for most tests:

- o Scanning source (src) interface: This is the interface from which test packets are sourced. This interface sources traffic to destination IPv6 addresses on the target network from a single link-local address, similar to how an adjacent intermediate node would transit traffic through the intermediate node.
- o Target network destination (dst) interface: This interface responds to neighbor solicitations as appropriate and confirms when an intermediate node has forwarded a packet to the interface for consumption. Where appropriate, the target network destination interface will respond to neighbor solicitations with a unique link-layer address per IPv6 address solicited.

#### 5. Modifiers (variables)

##### 5.1. Frequency of NDP triggering packets

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

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

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

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

## 6. Tests

### 6.1. Maximum number of valid hosts

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

#### 6.1.1. Test Streams

Two streams are defined:

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

#### 6.1.2. General Testing Procedure

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



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

2. Once the maximum rate for stream tester-new has been determined, transit packets for stream tester-new until 30 seconds have evolved. Then send packets matching the tester-renew stream every second. This specific step should be continued until packets in either stream don't reach the target network destination interface.
3. If all packets from the tester-renew stream don't reach the target network destination interface before the completion of the 2 minute test, reduce the rate of the tester-new stream and repeat the test until all packets in both streams are received by the target network interface.

#### 6.1.3. Discussion

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

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

#### 6.2. Stable-state time

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

##### 6.2.1. Test streams

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

##### 6.2.2. General Testing Procedure

1. Replicate behavior in "maximum number of valid hosts" until packets are not being received by the target network destination interface.
2. Back off the rate to a point just below where packets previously were not received by the target network interface.
3. Measure the duration in seconds required until all packets are consistently received by the target network interface.

#### 6.2.3. Discussion

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

#### 6.3. NDP Prioritization: Behavior with stale neighbor entries

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

##### 6.3.1. Test Streams

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

##### 6.3.2. General Testing Procedure

1. Send stream tester-new packets at a maximum rate as determined by "Maximum number of valid hosts."
2. Slow down stream tester-renew until one gets into refreshing every 6 seconds. If an address is in the "stale" state, it should get priority over new request.
3. Increase timer on stream tester-renew.
4. Stream tester-renew should always get responses. Stream tester-renew packets should always be received by the target network destination interface.
5. Stream tester-new should not always get responses. Stream tester-unreachable packets should not always be received by the target network destination interface.

### 6.3.3. Discussion

### 6.4. NDP Prioritization: Entries never present in neighbor cache

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

#### 6.4.1. Test Streams

#### 6.4.2. General Testing Procedure

#### 6.4.3. Discussion

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

### 6.5. NDP Prioritization: Unreachable addresses only

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

#### 6.5.1. Test Streams

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

#### 6.5.2. General Testing Procedure

1. Send stream tester-unreachable at a high rate for approximately 30 seconds, continuing traffic until the end of the test.
2. Send stream tester-new at a very low rate (perhaps once per second). Measure the rate at which the target network destination interface receives the packet.

#### 6.5.3. Discussion

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

## 7. Measurements Explicitly Excluded

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

### 7.1. DUT CPU Utilization

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

### 7.2. Malformed Packets

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

## 8. DUT initialization

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

## 9. IANA Considerations

This document makes no request of IANA.

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

## 10. Security Considerations

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

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

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

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

## 11. Acknowledgements

## 12. Normative References

- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, May 2008.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.

## Author's Address

Bill Cerveney  
Arbor Networks

Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: February 2014  
September 4, 2013

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

Traffic Management Benchmarking  
draft-constantine-bmwg-traffic-management-02.txt

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, 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 compare 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.....	4
1.1. Traffic Management Overview.....	4
1.2. DUT Lab Configuration and Testing Overview.....	5
2. Conventions used in this document.....	6
3. Scope and Goals.....	7
4. Traffic Benchmarking Metrics.....	7
4.1. Metrics for Stateless Traffic Tests.....	8
4.2. Metrics for Stateful Traffic Tests.....	9
5. Tester Capabilities.....	9
5.1. Stateless Test Traffic Generation.....	10
5.2. Stateful Test Pattern Generation.....	10
5.2.1. TCP Test Pattern Definitions.....	11
6. Traffic Benchmarking Methodology.....	12
6.1. Policing Tests.....	12
6.1.1 Policer Individual Tests.....	13
6.1.2 Policer Capacity Tests.....	14
6.1.2.1 Maximum Policers on Single Physical Port.....	15
6.1.2.2 Single Policer on All Physical Ports.....	15
6.1.2.3 Maximum Policers on All Physical Ports.....	15
6.2. Queue/Scheduler Tests.....	15
6.2.1 Queue/Scheduler Individual Tests.....	15
6.2.1.1 Testing Queue/Scheduler with Stateless Traffic....	15
6.2.1.2 Testing Queue/Scheduler with Stateful Traffic....	16
6.2.2 Queue / Scheduler Capacity Tests.....	17
6.2.2.1 Multiple Queues / Single Port Active.....	17
6.2.2.1.1 Strict Priority on Egress Port.....	17
6.2.2.1.2 Strict Priority + Weighted Fair Queue (WFQ)....	17
6.2.2.2 Single Queue per Port / All Ports Active.....	17
6.2.2.3 Multiple Queues per Port, All Ports Active.....	18
6.3. Shaper tests.....	18
6.3.1 Shaper Individual Tests.....	18
6.3.1.1 Testing Shaper with Stateless Traffic.....	18
6.3.1.2 Testing Shaper with Stateful Traffic.....	19
6.3.2 Shaper Capacity Tests.....	20
6.3.2.1 Single Queue Shaped, All Physical Ports Active....	20
6.3.2.2 All Queues Shaped, Single Port Active.....	20
6.3.2.3 All Queues Shaped, All Ports Active.....	20
6.4. Congestion Management tests.....	21
6.4.1 Congestion Management Verification Tests.....	21
6.4.1.1 Congestion Management with Stateless Traffic.....	21
6.4.1.2 Congestion Management with Stateful Traffic.....	21
6.4.2 Congestion Management Capacity Tests.....	22
6.4.2.1 All Data Queues with AQM, Single Physical Port...22	
6.4.2.1 All Data Queues with AQM, Multiple Physical Ports.22	
6.5. Concurrent Capacity Load Tests.....	23
7. Security Considerations.....	24
8. IANA Considerations.....	24
9. Conclusions.....	24
10. References.....	24
10.1. Normative References.....	24
10.2. Informative References.....	24
11. Acknowledgments.....	24



## 1. Introduction

Traffic management (i.e. policing, shaping, etc.) is an increasingly important component when engineering network Quality of Service (QoS) today. There is currently 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.

Specifically, this framework defines the methods to characterize the capacity of traffic management features in network devices, such as classification, policing, shaping, and active queue management.

This benchmarking framework can also be used as a test procedure to assist in the tuning of traffic management parameters before field deployment. In addition to Layer 2/3 benchmarking, Layer 4 test patterns are proposed by this draft in order to benchmark as close as possible to real end-user traffic.

### 1.1. Traffic Management Overview

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

- Traffic classification: identifies traffic according to various configuration rules (i.e. VLAN, DSCP, etc.) and marks this traffic internally to the network device. Multiple external priorities (DSCP, 802.1p, etc.) can map to the same priority in the device.
- Traffic policing: limits the rate of traffic that enters a network device according to the traffic classification. If the traffic exceeds the contracted limits, the traffic is either dropped or remarked and sent onto to the next network device
- Traffic Scheduling: provides traffic classification within the network device by directing packets to various types of queues and applies a dispatching algorithm to assign the forwarding sequence of packets
- Traffic shaping: a traffic control measure of actively buffering and metering the output rate in an attempt to adapt bursty traffic to the configured limits
- Active Queue Management (AQM): monitors the status of internal queues and actively drops (or re-marks) packets, which causes hosts using congestion-aware protocols 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.

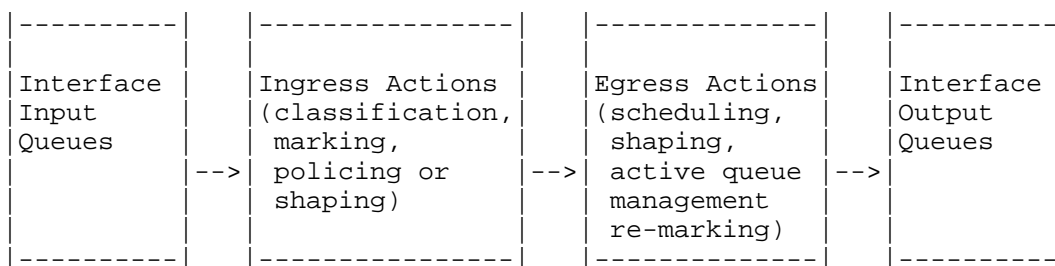


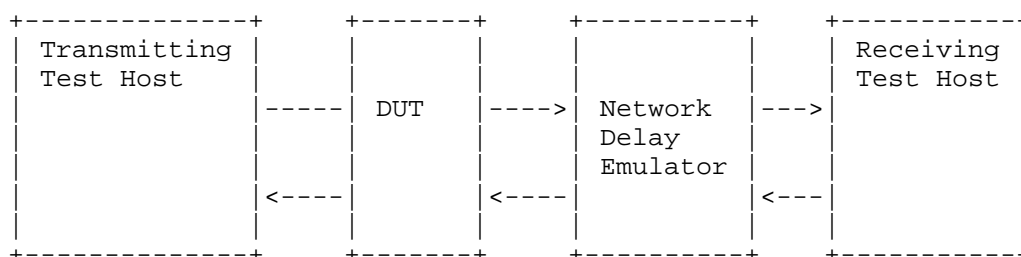
Figure 1: Generic Traffic Management capabilities of a Network Device

Ingress actions such classification are defined in RFC 4689 and include IP addresses, port numbers, DSCP, etc. In terms of marking, RFC 2697 and RFC 2698 define a single rate and dual rate, three color marker, respectively.

The MEF specifies policing and shaping in terms of Ingress and Egress Subscriber/Provider Conditioning Functions in MEF12.1; Ingress and Bandwidth Profile attributes in MEF 10.2 and MEF 26.

## 1.2 DUT Lab Configuration and Testing Overview

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.

This testing framework describes the individual tests and metrics for each of the following traffic management functions:

- Policing Tests
- Shaping Tests
- Queue / Scheduling Tests
- Congestion Management Tests

The tests are divided into individual tests and rated capacity tests. The individual tests are intended to verify the traffic management function according to the specifications. As an example, suppose a traffic shaper is to be tested at a CIR of 20 Mbps. The intent of the individual test is to test one instance of the shaper and it's ability to properly shape according to the metrics defined in section 4.

The capacity tests verify traffic management functions under full load. This involves concurrent testing of multiple interfaces with the specific traffic management function enabled, and doing so to the capacity limit of each interface.

For an example: a device is specified to be capable of shaping on all of it's egress ports. The individual test would first be conducted to benchmark the advertised shaping function against the metrics defined in section 4. Then the capacity test would be executed to test the shaping function concurrently on all interfaces and with maximum traffic load.

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. In other words, the DUT would be removed and the NDE performance benchmarked independently.

Note the NDE should be used in "full pipe" delay mode. Most NDEs allow for per flow delay actions, emulating QoS prioritization. For this framework, the NDE's sole purpose is simply to add delay to all packets (emulate network latency). So to benchmark the performance of the NDE, maximum offered load should be tested against the following frame sizes: 128, 256, 512, 768, 1024, 1500, and 9600 bytes. The delay accuracy at each of these packet sizes can then be used to calibrate the range of expected BDPS for the TCP stateful tests.

## 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:

BB: Bottleneck Bandwidth

BDP: Bandwidth Delay Product

BSA: Burst Size Achieved

CBS: Committed Burst Size

CIR: Committed Information Rate

DUT: Device Under Test

EBS: Excess Burst Size

EIR: Excess Information Rate

NDE: Network Delay Emulator

SP: Strict Priority Queuing

QL: Queue Length

QoS: Quality of Service

RED: Random Early Discard

RTT: Round Trip Time

SBS: Shaper Burst Size

SR: Shaper Rate

SSB: Send Socket Buffer

Tc: CBS Time Interval

Te: EBS Time Interval

Ti Transmission Interval

TTP: TCP Test Pattern

TTPET: TCP Test Pattern Execution Time

WRED: Weighted Random Early Discard

### 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
- General Layer 4-7 appliances (Proxies, WAN Accelerators, etc.)

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

The primary goal is to assess the maximum forwarding performance that a network device can sustain without dropping or impairing packets, or compromising the accuracy of multiple instances of traffic management functions. This is the benchmark for comparison between devices.

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 provides the test methods and metrics to conduct repeatable testing, which will provide the means to compare measured performance between DUTs.

As mentioned in section 1.2, this framework describes the individual tests and metrics for several management functions. It is also within scope that this framework will benchmark each function in terms of overall rated capacity. This involves concurrent testing of multiple interfaces with the specific traffic management function enabled, up to the capacity limit of each interface.

It is not within scope of this framework to specify the procedure for testing multiple traffic management functions concurrently. The multitudes of possible combinations is almost unbounded and the ability to identify functional "break points" would be most times impossible.

However, section 6.5 provides suggestions for some usual profiles of concurrent functions that would be useful to benchmark. The key requirement for any concurrent test function is that tests must produce reliable and repeatable results.

Also, it is not within scope to perform conformance testing. Tests defined in this framework benchmark the traffic management functions according to the metrics defined in section 4 and do not address any conformance to standards related to traffic management. Traffic management specifications largely do not exist and this is a prime driver for this framework; to provide an objective means to compare vendor traffic management functions.

Another goal is to devise methods that utilize flows with congestion-aware transport (TCP) as part of the traffic load and still produce repeatable results in the isolated test environment. This framework will 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 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 important to test along with stateless, UDP traffic in specific test scenarios (i.e. applications using TCP transport and UDP VoIP, etc.)

And finally, this framework will provide references to open source tools that can be used to provide stateless and/or stateful traffic generation emulation.

#### 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 segment layer metrics used for the stateful traffic testing.

##### 4.1. Metrics for Stateless Traffic Tests

For the stateless traffic tests, the metrics are defined at the layer 3 packet level versus layer 2 packet level for consistency.

Stateless traffic measurements require that sequence number and time-stamp be inserted into the payload for lost packet analysis. Delay analysis may be achieved by insertion of timestamps directly into the packets or timestamps stored elsewhere (packet captures). This framework does not specify the packet format to carry sequence number or timing information. However, RFC 4689 provides recommendations for sequence tracking along with definitions of in-sequence and out-of-order packets.

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 Excess 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 packets. As an example, the configured CBS of a DUT is 64KB and after the burst test, only a 63 KB can be achieved without packet loss. Then 63KB is the BSA. Also, the average Packet Delay Variation (PDV see below) is experienced by the packets sent at the BSA burst size should be recorded.
- Lost Packets (LP): For all traffic management tests, the tester will transmit the test packets into the DUT ingress port and the number of packets received at the egress port will be measured. The difference between packets transmitted into the ingress port and received at the egress port is the number of lost packets as measured at the egress port. These packets must have unique identifiers such that only the test packets are measured. RFC 4737 and RFC 2680 describe the need to establish the threshold to designate when a packet as lost, and the threshold MUST be reported with the results.
- Out of Sequence (OOS): in additions to LP metric, the test packets must be monitored for sequence and the out-of-sequence (OOS) packets. RFC 4689 defines the general function of sequence tracking, as well as definitions for in-sequence and out-of-order packets. Out-of-order packets will be counted per RFC 4737 and RFC 2680.
- Packet Delay (PD): the Packet Delay metric is the difference between the timestamp of the received egress port packets and the packets transmitted into the ingress port and specified in RFC 2285.

- Packet Delay Variation (PDV): the Packet Delay Variation metric is the variation between the timestamp of the received egress port packets and specified in RFC 5481.

#### 4.2. Metrics for Stateful Traffic Tests

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

- TCP Test Pattern Execution Time (TTPET): 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 include bulk transfer and interactive applications. The interactive patterns include application models such as HTTP business applications, database applications, etc. The TTPET will be the measure of the time for a single execution of a TCP Test Pattern (TTP). Average, minimum, and maximum times will be measured or calculated.

An example would be an interactive HTTP TTP session which should take 5 seconds on a GigE network with 0.5 msec latency. During ten (10) executions of this TTP, the TTPER results might be: average of 6.5 seconds, minimum of 5.0 seconds, and maximum of 7.9 seconds.

- 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. These retransmitted bytes should be recorded from the sender's TCP/IP stack perspective, to avoid any misinterpretation that a reordered packet is a retransmitted packet (as may be the case with packet decode interpretation).



- Buffer Delay: represents the increase in RTT during a TCP test versus the baseline DUT RTT (non congested, inherent latency). RTT and the technique to measure RTT (average versus baseline) are defined in RFC 6349. Referencing RFC 6349, the average RTT is derived from the total of all measured RTTs during the actual test sampled at every second divided by the test duration in seconds.

$$\text{Average RTT during transfer} = \frac{\text{Total RTTs during transfer}}{\text{Transfer duration in seconds}}$$

$$\text{Buffer Delay \%} = \frac{\text{Average RTT during Transfer} - \text{Baseline RTT}}{\text{Baseline RTT}} \times 100$$

Note that even though this was not explicitly stated in RFC 6349, retransmitted packets should not be used in RTT measurements.

Also, the test results should record the average RTT in msec across the entire test duration and number of samples.

## 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 traffic at up to the link speed of the DUT. The test set must be calibrated to verify that it will not drop any packets. The test set's inherent PD and PDV must also be calibrated and subtracted from the PD and PDV metrics. The test set must support the encapsulation to be tested such as VLAN, Q-in-Q, MPLS, etc. Also, the test set must allow control of the classification techniques defined in RFC 4689 (i.e. IP address, DSCP, TOS, etc classification).

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 (e.g. GigE, 10 GigE, etc.). 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 device 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). In some cases, scaled down TCP implementations can also be used as is sometimes the case for high performance, hardware-based commercial implementations.

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 bi-directional and can be chatty.

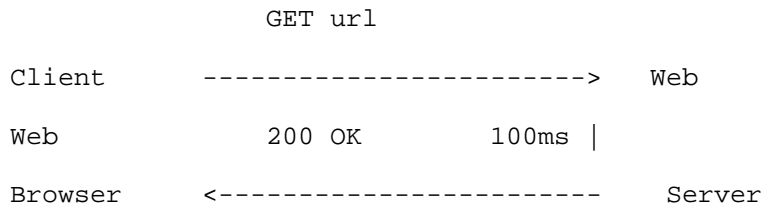
The test device must not only support bulk TCP transfer application traffic but also chatty traffic. A valid stress test SHOULD include both traffic types. This is due to the non-uniform, bursty nature of chatty applications versus the relatively uniform nature of bulk transfers (the bulk transfer smoothly stabilizes to equilibrium state under lossless conditions).

While iperf is an excellent choice for TCP bulk transfer testing, the open source tool "Flowgrind" (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 (e.g Gig, 10 GigE).

#### 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 why stateful TCP test traffic must be used.

An application could be fully emulated up to Layer 7, however this framework proposes that stateful TCP test patterns be used in order 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 content to the Client (after a Server delay of 100 msec). This asynchronous, "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. These examples reflect those specified in "draft-ietf-bmwg-ca-bench-meth-04" which suggests traffic mixes for a variety of representative application profiles.

There are two (2) techniques recommended by this framework to develop standard TCP test patterns for traffic management benchmarking.

The first technique involves modeling, 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 on 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 techniques.

(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.

Each test should compare the network device's internal statistics (available via command line management interface, SNMP, etc.) to the measured metrics defined in section 4. This evaluates the accuracy of the internal traffic management counters under verification test conditions and capacity test conditions that are defined in each subsection.

### 6.1. Policing Tests

The intent of the policing tests is to verify the policer performance (i.e. CIR-CBS and EIR-EBS parameters). The tests will verify that the network device can handle the CIR with CBS and the EIR with EBS and will use back-back packet 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.

The tests are divided into two (2) sections; individual policer function verification tests and then full capacity policing tests. It is important to verify the basic functionality of the individual policer then proceed into the fully rated capacity of the device. This capacity may include the number of policing policies per device and the number of policers simultaneously active across all ports.

#### 6.1.1 Policer Individual Tests

Policing tests should use stateless traffic. Stateful TCP test traffic will generally be adversely affected by a policer in the absence of traffic shaping. So while TCP traffic could be used, it is more accurate to benchmark a policer with stateless traffic.

The policer test shall test a policer as defined by RFC 4115 or MEF 10.2, depending upon the equipment's specification. As an example for RFC 4115, consider a CBS and EBS of 64KB and CIR and EIR of 100 Mbps on a 1GigE physical link (in color-blind mode). A stateless traffic burst of 64KB would be sent into the policer at the GigE rate. This equates to approximately a 0.512 msec burst time and the burst-to-burst spacing would be 5.12 msec.

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 packets 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 packets are dropped.

Additional tests beyond the simple color-blind example might include: color-aware mode, configurations where EIR is greater than CIR, etc.

#### 6.1.2 Policer Capacity Tests

The intent of the capacity tests is to verify the policer performance in a scaled environment with multiple ingress customer policers on multiple physical ports. This test will benchmark the maximum number of active policers as specified by the device manufacturer.

As an example, a Layer 2 switching device may specify that each of the 32 physical ports can be policed using a pool of policing service policies. The device may carry a single customer's traffic on each physical port and a single policer is instantiated per physical port. Another possibility is that a single physical port may carry multiple customers, in which case many customer flows would be policed concurrently on an individual physical port.

The specified policing function capacity is generally expressed in terms of the number of policers active on each individual physical port as well as the number of unique policer rates that are utilized. For all of the capacity tests, the benchmarking methodology described in Section 6.1.1 for a single policer should be applied to each of the physical port policers.

##### 6.1.2.1 Maximum Policers on Single Physical Port

The first policer capacity test will benchmark a single physical port, maximum policers on that physical port.

Assume multiple categories of ingress policers at rates  $r_1, r_2, \dots, r_n$ . There are multiple customers on a single physical port. Each customer could be represented by a single tagged vlan, double tagged vlan, VPLS instance etc. Each customer is mapped to a different policer. Each of the policers can be of rates  $r_1, r_2, \dots, r_n$ . Policer granularity guideline (do we need ??)

An example configuration would be

- Y1 customers, policer rate  $r_1$
- Y2 customers, policer rate  $r_2$
- Y3 customers, policer rate  $r_3$
- ...
- Yn customers, policer rate  $r_n$

Some bandwidth on the physical port is dedicated for other traffic (non customer traffic); this includes network control protocol traffic. There is a separate policer for the other traffic. Typical deployments have 3 categories of policers; there may be some deployments with more or less than 3 categories of ingress policers.



#### 6.1.2.2 Single Policer on All Physical Ports

The second policer capacity test involves a single Policer function per physical port with all physical ports active. In this test, there is a single policer per physical port. The policer can have one of the rates  $r_1, r_2, \dots, r_n$ . All the physical ports in the networking device are active.

#### 6.1.2.3 Maximum Policers on All Physical Ports

Finally the third policer capacity test involves a combination of the first and second capacity test, namely maximum policers active per physical port and all physical ports are active .

### 6.2. Queue and Scheduler Tests

Queues and traffic Scheduling are closely related in that a queue's priority dictates the manner in which the traffic scheduler's transmits packets out of the egress port.

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).

Similar to the policing tests, the tests are divided into two sections; individual queue/scheduler function verification tests and then full capacity tests.

#### 6.2.1 Queue/Scheduler Individual Tests

The various types of scheduling techniques include FIFO, Strict Priority (SP), Weighted Fair Queueing (WFQ) along with other variations. This test framework recommends to test at a minimum these three techniques although it is the discretion of the tester to benchmark other device scheduling algorithms.

##### 6.2.1.1 Testing Queue/Scheduler 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 ( $T_i$ ) 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  $T_c / T_e$  time interval discussed in the policer section 6.1 and is as follows:

$$T_i = QL * 8 / BB$$

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  $T_i$  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 packets are dropped.

The scheduling function must also be characterized during the test to benchmark the device's ability to schedule the queues according to the priority. An example would be 2 levels of priority including SP and FIFO queueing. Under flow load greater than the egress port speed, the higher priority packets should be transmitted without drops (and also maintain low latency), while the lower priority (or best effort) queue may be dropped.

#### 6.2.1.2 Testing Queue/Scheduler 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. Referencing RFC6349, 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.

### 6.2.2 Queue / Scheduler Capacity Tests

The intent of these capacity tests is to verify queue/scheduler performance in a scaled environment with multiple queues/schedulers active on multiple egress physical ports. This test will benchmark the maximum number of queues and schedulers as specified by the device manufacturer. Each priority in the system will map to a separate queue.

#### 6.2.2.1 Multiple Queues / Single Port Active

For the first scheduler / queue capacity test, multiple queues per port will be tested on a single physical port. In this case, all the queues (typically 8) are active on a single physical port. Traffic from multiple ingress physical ports are directed to the same egress physical port which will cause oversubscription on the egress physical port.

There are many types of priority schemes and combinations of priorities that are managed by the scheduler. The following sections specify the priority schemes that should be tested.

##### 6.2.2.1.1 Strict Priority on Egress Port

For this test, Strict Priority (SP) scheduling on the egress physical port should be tested and the benchmarking methodology specified in section 6.2.1 should be applied here. For a given priority, each ingress physical port should get a fair share of the egress physical port bandwidth.

##### 6.2.2.1.2 Strict Priority + Weighted Fair Queue (WFQ) on Egress Port

For this test, Strict Priority (SP) and Weighted Fair Queue (WFQ) should be enabled simultaneously in the scheduler but on a single egress port. The benchmarking methodology specified in Section 6.2.1 should be applied here. Additionally, the egress port bandwidth sharing among weighted queues should be proportional to the assigned weights. For a given priority, each ingress physical port should get a fair share of the egress physical port bandwidth.

#### 6.2.2.2 Single Queue per Port / All Ports Active

Traffic from multiple ingress physical ports are directed to the same egress physical port, which will cause oversubscription on the egress physical port. Also, the same amount of traffic is directed to each egress physical port.

The benchmarking methodology specified in Section 6.2.1 should be applied here. Each ingress physical port should get a fair share of the egress physical port bandwidth. Additionally, each egress physical port should receive the same amount of traffic.

#### 6.2.2.3 Multiple Queues per Port, All Ports Active

Traffic from multiple ingress physical ports are directed to all queues of each egress physical port, which will cause oversubscription on the egress physical ports. Also, the same amount of traffic is directed to each egress physical port.

The benchmarking methodology specified in Section 6.2.1 should be applied here. For a given priority, each ingress physical port should get a fair share of the egress physical port bandwidth. Additionally, each egress physical port should receive the same amount of traffic.

### 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 framework will discuss testing at the egress (although the technique can be applied to ingress and internal queues).

Similar to the policing tests, the tests are divided into two sections; individual shaper function verification tests and then full capacity shaper tests.

#### 6.3.1 Shaper Individual Tests

A network device's traffic shaper will generally either shape to an average rate or provide settings similar to a policer (e.g. 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  $T_c$  for the policer which is indicated in section 6.1.

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

##### 6.3.1.1 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 interface.

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

#### 6.3.1.2 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.

The cumulative TCP Window Sizes\* (RWND at the receiving end & CWND at the transmitting end) equates to:

TCP window size\* for each connection x number of connections

\* as described in section 3 of RFC6349, the SSB MUST be large enough to fill the BDP

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 types may include: Small Web Site, Email, Citrix, etc.

The test results will be recorded per the stateful metrics defined in section 4.2, primarily the TCP Test Pattern Execution Time (TPPET), 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 interface
- Variation (min, max) around the shaper rate

#### 6.3.2 Shaper Capacity Tests

The intent of these scalability tests is to verify shaper performance in a scaled environment with shapers active on multiple queues on multiple egress physical ports. This test will benchmark the maximum number of shapers as specified by the device manufacturer.

For all of the capacity tests, the benchmarking methodology described in Section 6.3.1 for a single shaper should be applied to each of the physical port and/or queue shapers.

##### 6.3.2.1 Single Queue Shaped, All Physical Ports Active

The first shaper capacity test involves per port shaping, all physical ports active. Traffic from multiple ingress physical ports are directed to the same egress physical port and this will cause oversubscription on the egress physical port. Also, the same amount of traffic is directed to each egress physical port.

The benchmarking methodology described in Section 6.3.1 should be applied to each of the physical ports. Each ingress physical port should get a fair share of the egress physical port bandwidth.

##### 6.3.2.2 All Queues Shaped, Single Port Active

The second shaper capacity test is conducted with all queues actively shaping on a single physical port. The benchmarking methodology described in per port shaping test (previous section) serves as the foundation for this. Additionally, each of the SP queues on the egress physical port is configured with a shaper. For the highest priority queue, the maximum amount of bandwidth available is limited by the bandwidth of the shaper. For the lower priority queues, the maximum amount of bandwidth available is limited by the bandwidth of the shaper and traffic in higher priority queues.

##### 6.3.2.3 All Queues Shaped, All Ports Active

And for the third shaper capacity test (which is a combination of the tests in the previous two sections), all queues will be actively shaping and all physical ports active.

#### 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 main goal is to discard traffic before the queue overflows as is the case for a FIFO queue. This discard in effect sends implicit 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).

Similar to the policing tests, the tests are divided into two (2) sections; individual AQM function verification tests and then full capacity AQM tests.

##### 6.4.1 Congestion Management Verification Tests

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](http://en.wikipedia.org/wiki/Random_early_detection)). In some network devices, this discard threshold is discretely configurable (e.g. 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.

###### 6.4.1.1. 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 results 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)

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

#### 6.4.2 Congestion Management Capacity Tests (TBD)

Only for the data queues (bursty traffic), different AQM techniques = RED, WRED, etc.

##### 6.4.2.1 All Data Queues with AQM, Single Physical Port

TBD

##### 6.4.2.1 All Data Queues with AQM, Multiple Physical Ports

TBD

#### 6.5 Concurrent Capacity Load Tests

As mentioned in the scope of this document, it is impossible to specify the various permutations of concurrent traffic management functions that should be tested in a device for capacity testing. However, some profiles are listed below which may be useful to test under capacity as well:

- Policers on ingress and queuing on egress
- Policers on ingress and shapers on egress (not intended for a flow to be policed then shaped, these would be two different flows tested at the same time)
- etc

#### Appendix A: Open Source Tools for Traffic Management Testing

This framework specifies that stateless and stateful behaviors should both be tested. 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, packet size, UDP port, and test duration can be controlled. A report of bytes transmitted, packets 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 packet 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 packet loss and within the packet 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 specify repeatable TCP test patterns for traffic management benchmarking and provides examples of the following test patterns:

- Bulk: generate concurrent TCP connections whose aggregate number of in-flight data bytes would fill the BDP. Guidelines from RFC 6349 are used to create this traffic model.
- Bursty: generate precise burst patterns within a single or multiple TCP session(s). The idea is for TCP to establish equilibrium and then burst application bytes at defined sizes.
- AQM: generate various burst sizes within a TCP session, spacing the bursts apart such that 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 from Flowgrind to generate this TCP test pattern.
- Cirix: mimic 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.

#### 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

### Authors' Addresses

Barry Constantine

JDSU, Test and Measurement Division

Germantown, MD 20876-7100, USA

Phone: +1 240 404 2227

Email: [barry.constantine@jdsu.com](mailto:barry.constantine@jdsu.com)

Timothy Copley

Level 3 Communications

14605 S 50th Street

Phoenix, AZ 85044

Email: [Timothy.copley@level3.com](mailto:Timothy.copley@level3.com)

Ram Krishnan

Brocade Communications

San Jose, 95134, USA

Phone: +001-408-406-7890

Email: [ramk@brocade.com](mailto:ramk@brocade.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 10, 2014

L. Avramov  
Cisco Systems  
J. Rapp  
Hewlett Packard  
November 6, 2013

Definitions and Metrics for Data Center Benchmarking  
draft-dcbench-def-01

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

This Internet-Draft will expire on May 10, 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 . . . . .	4
1.2. Definition format . . . . .	4
2. Latency . . . . .	4
2.1. Definition . . . . .	4
2.2 Discussion . . . . .	5
2.3 Measurement Units . . . . .	5
3 Jitter . . . . .	6
3.1 Definition . . . . .	6
3.2 Discussion . . . . .	6
3.3 Measurement Units . . . . .	6
4 Physical Layer Calibration . . . . .	6
4.1 Definition . . . . .	6
4.2 Discussion . . . . .	7
4.3 Measurement Units . . . . .	7
5 Line rate . . . . .	7
5.1 Definition . . . . .	7
5.2 Discussion . . . . .	8
5.3 Measurement Units . . . . .	9
6 Buffering . . . . .	10
6.1 Buffer . . . . .	10
6.1.1 Definition . . . . .	10
6.1.3 Discussion . . . . .	11
6.1.3 Measurement Units . . . . .	11
6.2 Incast . . . . .	12
6.2.1 Definition . . . . .	12
6.2.2 Discussion . . . . .	12
6.2.3 Measurement Units . . . . .	13
7 Application Throughput: Data Center Goodput . . . . .	13
7.1. Definition . . . . .	13
7.2. Discussion . . . . .	13
7.3. Measurement Units . . . . .	13
8. References . . . . .	14
3.1. Normative References . . . . .	14
3.2. Informative References . . . . .	14
3.3. URL References . . . . .	14

3.4. Acknowledgments . . . . .	14
Authors' Addresses . . . . .	15

## 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 Units: 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.

The Latency interval can be assessed between different combinations of events, irrespectively of the type of switching device (bit forwarding aka cut-through or store forward type of device)

Traditionally the latency measurement definitions are:

FIFO (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

LIFO (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.

Another possibility to summarize the four different definitions above is to refer to the bit position as they normally occur: input to output.

FIFO is FL (First bit Last bit)	FIFO is FF (First bit First bit)
LILO is LL (Last bit Last bit)	LIFO is LF (Last bit First bit)

This definition explained in this section in context of data center switching benchmarking is in lieu of 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

FIFO is the most important measuring definition. Any type of switches MUST be measured with the FIFO mechanism: FIFO 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 FIFO.

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 Units

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

1) FIFO 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.

Even with the reference to RFC 3393, there are many definitions of "jitter" possible. The one selected for Data Center Benchmarking is closest to RFC 3393.

#### 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 Units

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 Units

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 equipment].

### 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 +/- 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 Units

"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. -The PPM

adjustment SHOULD only be used for a line rate type of measurement

## 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. (please refer to the buffer efficiency section for further details).

**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 Units

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 Units

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 Units

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

### 3.1. 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.

### 3.2. 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.

### 3.3. 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>

### 3.4. Acknowledgments

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



Authors' Addresses

Jacob Rapp  
Hewlett-Packard Company  
3000 Hanover Street  
Palo Alto, CA 94304  
United States  
Phone: +1 650 857 3367  
Email: jacob.h.rapp@hp.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.

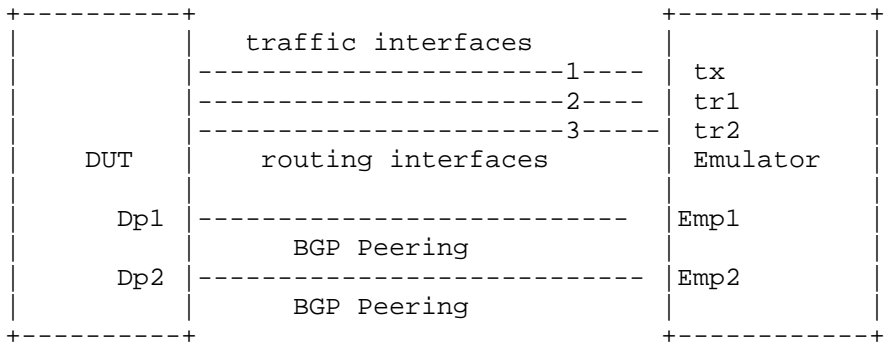


Figure 1 Basic Test Setup

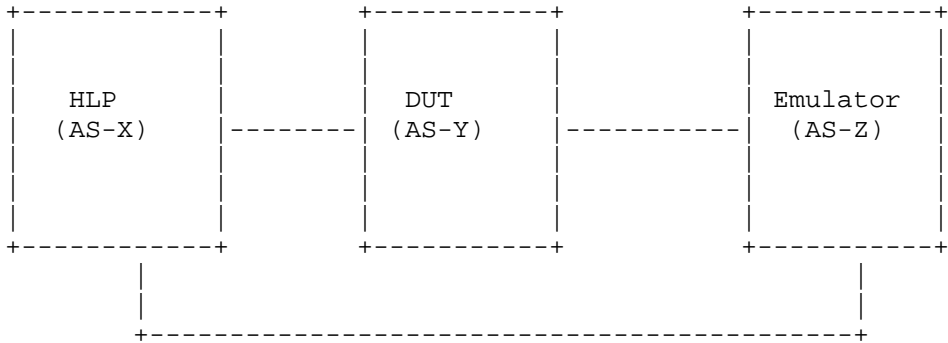


Figure 2 Three Node Setup for eBGP and iBGP Convergence



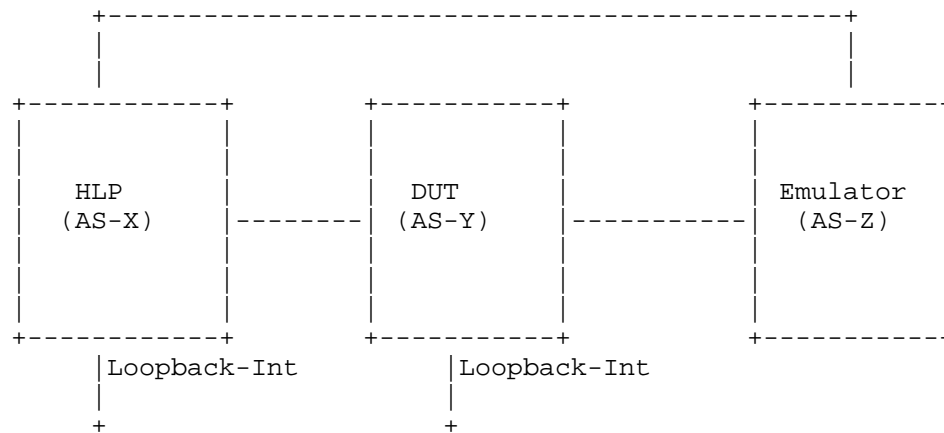


Figure 3 BGP Convergence for eBGP Multihop Scenario

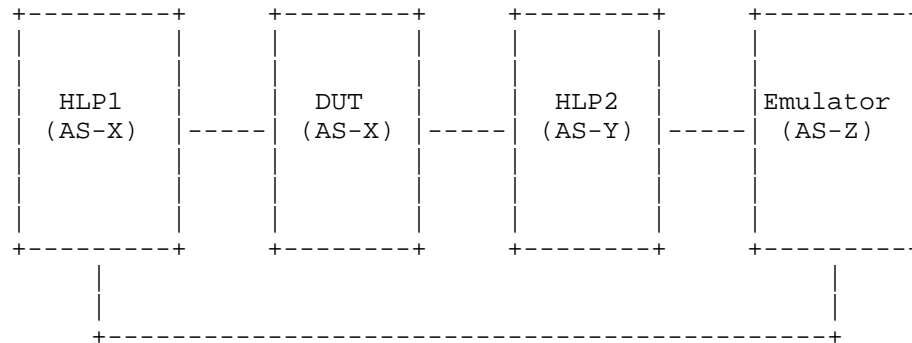


Figure 4 Four Node Setup for EBGP and IBGP Convergence

#### 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  $Tup(EMx, Rt-A)$  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

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

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

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

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

- 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](mailto:rajiv.papneja@huawei.com)

Bhavani Parise  
Cisco Systems

Email: [bhavani@cisco.com](mailto:bhavani@cisco.com)

Susan Hares  
Adara Networks

Email: [shares@ndzh.com](mailto:shares@ndzh.com)

Dean Lee  
IXIA

Email: [dlee@ixiacom.com](mailto:dlee@ixiacom.com)

Ilya Varlashkin  
Easynet Global Services

Email: [ilya.varlashkin@easynet.com](mailto:ilya.varlashkin@easynet.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 label

## 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

