

IPPM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 4, 2014

A. Akhter  
B. Claise  
Cisco Systems, Inc.  
March 3, 2014

Passive Performance Metrics Sub-Registry  
draft-akhter-ippm-registry-passive-01.txt

Abstract

This memo defines the Passive Performance Metrics sub-registry of the Performance Metric Registry. This sub-registry will contain Passive Performance Metrics, especially those defined in RFCs prepared in the IP Performance Metrics (IPPM) Working Group of the IETF, and possibly applicable to other IETF metrics.

IPPM Passive metric registration is meant to allow wider adoption of common metrics in an inter-operable way. There are challenges with metric interoperability and adoption (to name a few) due to flexible input parameters, confusion between many similar metrics, and varying output formats.

This memo proposes a way to organize registry entries into columns that are well-defined, permitting consistent development of entries over time (a column may be marked NA if it is not applicable for that metric). The design is intended to foster development of registry entries based on existing reference RFCs, whilst each column serves as a check-list item to avoid omissions during the registration process. Every entry in the registry, before IANA action, requires Expert review as defined by concurrent IETF work in progress "Registry for Performance Metrics" (draft-manyfolks-ippm-metric-registry).

The document contains example entries for the Passive Performance Metrics sub-registry: a registry entry for a passive metric based on octetTotalCount as defined in RFC5102 and a protocol specific passive metric based on RTP packets lost as defined in RFC3550. The examples are for Informational purposes and do not create any entry in the IANA registry.

Requirements Language

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

## Status of This Memo

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

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

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

This Internet-Draft will expire on September 4, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
2. Background and Motivation: . . . . .	6
3. Scope . . . . .	6
4. Passive Registry Categories and Columns . . . . .	7
4.1. Common Registry Indexes and Information . . . . .	7
4.1.1. Identifier . . . . .	7
4.1.2. Name . . . . .	7
4.1.3. Status . . . . .	7
4.1.4. Requester . . . . .	7
4.1.5. Revision . . . . .	7
4.1.6. Revision Date . . . . .	7
4.1.7. Description . . . . .	7
4.1.8. Reference Specification(s) . . . . .	8
4.2. Metric Definition . . . . .	8

4.2.1.	Reference Definition . . . . .	8
4.2.2.	Fixed Parameters . . . . .	8
4.3.	Method of Measurement . . . . .	8
4.3.1.	Reference Implementation . . . . .	8
4.3.2.	Traffic Filter Criteria . . . . .	9
4.3.3.	Measurement Timing . . . . .	9
4.3.4.	Output Type(s) and Data Format . . . . .	9
4.3.5.	Metric Units . . . . .	10
4.3.6.	Run-time Parameters and Data Format . . . . .	10
4.4.	Comments and Remarks . . . . .	10
5.	Example Generalized Passive Octet Count Entry . . . . .	11
5.1.	Registry Indexes . . . . .	11
5.1.1.	Element Identifier . . . . .	11
5.1.2.	Metric Name . . . . .	11
5.1.3.	Status . . . . .	11
5.1.4.	Requester . . . . .	11
5.1.5.	Revision . . . . .	11
5.1.6.	Revision Date . . . . .	11
5.1.7.	Metric Description . . . . .	12
5.1.8.	Reference Specification(s) . . . . .	12
5.2.	Metric Definition . . . . .	12
5.2.1.	Reference Definition . . . . .	12
5.2.2.	Fixed Parameters . . . . .	12
5.3.	Method of Measurement . . . . .	12
5.3.1.	Reference Implementation . . . . .	12
5.3.2.	Traffic Filter Criteria . . . . .	12
5.3.3.	Measurement Timing . . . . .	12
5.3.4.	Output Type(s) and Data Format . . . . .	13
5.3.5.	Metric Units . . . . .	13
5.3.6.	Run-time Parameters and Data Format . . . . .	13
5.4.	Comments and Remarks . . . . .	13
6.	Example 5min Passive Egress Octet Count Entry on WAN Interface . . . . .	13
6.1.	Registry Indexes . . . . .	14
6.1.1.	Element Identifier . . . . .	14
6.1.2.	Metric Name . . . . .	14
6.1.3.	Status . . . . .	14
6.1.4.	Requester . . . . .	14
6.1.5.	Revision . . . . .	14
6.1.6.	Revision Date . . . . .	14
6.1.7.	Metric Description . . . . .	14
6.1.8.	Reference Specification(s) . . . . .	15
6.2.	Metric Definition . . . . .	15
6.2.1.	Reference Definition . . . . .	15
6.2.2.	Fixed Parameters . . . . .	15
6.3.	Method of Measurement . . . . .	15
6.3.1.	Reference Implementation . . . . .	15
6.3.2.	Traffic Filter Criteria . . . . .	15

6.3.3.	Measurement Timing . . . . .	16
6.3.4.	Output Type(s) and Data Format . . . . .	16
6.3.5.	Metric Units . . . . .	16
6.3.6.	Run-time Parameters and Data Format . . . . .	16
6.4.	Comments and Remarks . . . . .	16
7.	Example Passive RTP Lost Packet Count . . . . .	16
8.	Example BLANK Registry Entry . . . . .	16
8.1.	Registry Indexes . . . . .	17
8.1.1.	Element Identifier . . . . .	17
8.1.2.	Metric Name . . . . .	17
8.1.3.	Status . . . . .	17
8.1.4.	Requester . . . . .	17
8.1.5.	Revision . . . . .	17
8.1.6.	Revision Date . . . . .	17
8.1.7.	Metric Description . . . . .	17
8.1.8.	Reference Specification(s) . . . . .	17
8.2.	Metric Definition . . . . .	17
8.2.1.	Reference Definition . . . . .	17
8.2.2.	Fixed Parameters . . . . .	18
8.3.	Method of Measurement . . . . .	18
8.3.1.	Reference Implementation . . . . .	18
8.3.2.	Traffic Filter Criteria . . . . .	18
8.3.3.	Measurement Timing . . . . .	18
8.3.4.	Output Type(s) and Data Format . . . . .	18
8.3.5.	Metric Units . . . . .	18
8.3.6.	Run-time Parameters and Data Format . . . . .	18
8.4.	Comments and Remarks . . . . .	19
9.	Security Considerations . . . . .	19
10.	IANA Considerations . . . . .	19
11.	Acknowledgements . . . . .	20
12.	References . . . . .	20
12.1.	Normative References . . . . .	20
12.2.	Informative References . . . . .	20
	Authors' Addresses . . . . .	21

## 1. Introduction

The IETF has been specifying and continues to specify Performance Metrics. While IP Performance Metrics (IPPM) is the working group (WG) primarily focusing on Performance Metrics definition at the IETF, other working groups, have also specified Performance Metrics:

The "Metric Blocks for use with RTCP's Extended Report Framework" [XRBLOCK] WG recently specified many Performance Metrics related to "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611], which establishes a framework to allow new information to be conveyed in RTCP, supplementing the original report blocks defined

in "RTP: A Transport Protocol for Real-Time Applications", [RFC3550].

The Benchmarking Methodology" [BMWG] WG proposed some Performance Metrics as part of the benchmarking methodology.

The IP Flow Information eXport WG (IPFIX) [IPFIX] has existing and proposed Information Elements related to performance metrics.

The Performance Metrics for Other Layers (PMOL) [PMOL], a concluded working group, defined some Performance Metrics related to Session Initiation Protocol (SIP) voice quality [RFC6035], as well as guidelines for defining performance metrics [RFC6390]

It is expected that more and more Performance Metrics will be defined in the future, not only IP based metrics, but also protocol-specific ones and application-specific ones.

However, there is currently no Performance Metrics registry in IANA. "Registry for Performance Metrics" [I-D.manyfolks-ippm-metric-registry] defines a common registry for metrics. The registry proposes the creation of two sub-registries, one for active metrics and another for passive measurements.

There is a sister document for the active metric sub-registry in "Active Performance Metric Sub-Registry" [I-D.mornuley-ippm-registry-active].

This document defines the Passive Performance Measurements Sub-Registry of the Performance Metric Registry. This sub-registry will contain passive performance metrics that meet the criteria set by the IETF and review of the Performance Metric Experts. It is expected that the majority of the metrics will have been defined elsewhere within the IETF working groups such as IPPM, BMWG, IPFIX, etc.

This sub-registry is part of the Performance Metric Registry [I-D.manyfolks-ippm-metric-registry] which specifies that all sub-registries must contain at least the following common fields: the identifier, the name, the status, the requester, the revision, the revision date, the description for each entry, and the reference specifications used as the foundation for the Registered Performance Metric (see [I-D.manyfolks-ippm-metric-registry]). In addition to these common fields the passive metrics sub-registry has additional fields that provide the necessary background for interoperability and adoption.

## 2. Background and Motivation:

(from draft-mornuley-ippm-registry-active):

One clear motivation for having such a registry is to allow a controller to request a measurement agent to perform a measurement using a specific metric (see [I-D.ietf-lmap-framework]). Such a request can be performed using any control protocol that refers to the value assigned to the specific metric in the registry. Similarly, the measurement agent can report the results of the measurement and by referring to the metric value it can unequivocally identify the metric that the results correspond to.

There are several side benefits of having a registry with well-chosen entries. First, the registry could serve as an inventory of useful and used metrics that are normally supported by different implementations of measurement agents. Second, the results of the metrics would be comparable even if they are performed by different implementations and in different networks, as the metric and method is unambiguously defined.

## 3. Scope

[I-D.manyfolks-ippm-metric-registry] defines the overall structure for a Performance Metric Registry and provides guidance for defining a sub registry.

This document defines the Passive Performance Metrics Sub-registry; passive metrics are those where the measurements are based the observation of on user traffic. Specifically, this traffic has not been generated for the purpose of measurement.

A row in the registry corresponds to one Registered Performance Metric, with entries in the various columns specifying the metric. Section 4 defines the additional columns for a Registered Passive Performance Metric.

As discussed in [I-D.manyfolks-ippm-metric-registry], each entry (row) must be tightly defined; the definition must leave open only a few parameters that do not change the fundamental nature of the measurement (such as source and destination addresses), and so promotes comparable results across independent implementations. Also, each registered entry must be based on existing reference RFCs (or other standards) for performance metrics, and must be operationally useful and have significant industry interest. This is ensured by expert review for every entry before IANA action.

#### 4. Passive Registry Categories and Columns

This section defines the categories and columns of the registry. Below, categories are described at the 4.x heading level, and columns are at the 4.x.y heading level. There are three categories, divided into common information (from [I-D.manyfolks-ippm-metric-registry]), metric definition and an open Comments section.

##### 4.1. Common Registry Indexes and Information

This category has multiple indexes to each registry entry. It is defined in [I-D.manyfolks-ippm-metric-registry]:

###### 4.1.1. Identifier

Defined in [I-D.manyfolks-ippm-metric-registry]. Definition text to be copied once source is stable.

###### 4.1.2. Name

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as above.

###### 4.1.3. Status

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as above.

###### 4.1.4. Requester

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as above.

###### 4.1.5. Revision

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as above.

###### 4.1.6. Revision Date

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as above.

###### 4.1.7. Description

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as the above.

#### 4.1.8. Reference Specification(s)

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as the above.

#### 4.2. Metric Definition

This category includes columns to prompt all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters, which are left open in the origin definition but have a particular value defined by the performance metric.

##### 4.2.1. Reference Definition

This entry provides references to relevant sections of the RFC(s) defining the metric, as well as any supplemental information needed to ensure an unambiguous definition for implementations.

##### 4.2.2. Fixed Parameters

Fixed Parameters are input factors whose value must be specified in the Registry. The measurement system uses these values.

Where referenced metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Fixed Parameters. For example, for RTP packet loss calculation relies on the validation of a packet as RTP which is a multi-packet validation controlled by MIN\_SEQUENTIAL as defined by [RFC3550]. Varying MIN\_SEQUENTIAL values can alter the loss report and this value could be set as a fixed parameter.

A Parameter which is Fixed for one Registry entry may be designated as a Run-time Parameter for another Registry entry.

#### 4.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

##### 4.3.1. Reference Implementation

This entry provides references to relevant sections of the RFC(s) describing the method of measurement, as well as any supplemental information needed to ensure unambiguous interpretation for implementations referring to the RFC text.



Specifically, this section should include pointers to pseudocode or actual code that could be used for an unambiguous implementation.

#### 4.3.2. Traffic Filter Criteria

The filter specifies the traffic constraints that the passive measurement method used is valid (or invalid) for. This includes valid packet sampling ranges, width of valid traffic matches (eg. all traffic on interface, UDP packets in a flow (eg. same RTP session)).

It is possible that the measurement method may not have a specific limitation. However, this specific registry entry with it's combination of fixed parameters implies restrictions. These restrictions would be listed in this field.

#### 4.3.3. Measurement Timing

Measurement timing defines the behavior of the measurement method with respect to timing.

Is the measurement continuous?

If the measurement is sampled, what is the format of sampling? (eg random packet, random time, etc.)

How long is the measurement interval?

#### 4.3.4. Output Type(s) and Data Format

For entries which involve a stream and many singleton measurements, a statistic may be specified in this column to summarize the results to a single value. If the complete set of measured singletons is output, this will be specified here.

Some metrics embed one specific statistic in the reference metric definition, while others allow several output types or statistics.

Each entry in the output type column contains the following information:

- o Value: The name of the output type
- o Data Format: provided to simplify the communication with collection systems and implementation of measurement devices.
- o Reference: the specification where the output type is defined

The output type defines the type of result that the metric produces. It can be the raw result(s) or it can be some form of statistic. The specification of the output type must define the format of the output. In some systems, format specifications will simplify both measurement implementation and collection/storage tasks. Note that if two different statistics are required from a single measurement (for example, both "Xth percentile mean" and "Raw"), then a new output type must be defined ("Xth percentile mean AND Raw").

#### 4.3.5. Metric Units

The measured results must be expressed using some standard dimension or units of measure. This column provides the units.

When a sample of singletons (see [RFC2330] for definitions of these terms) is collected, this entry will specify the units for each measured value.

#### 4.3.6. Run-time Parameters and Data Format

Run-Time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete. However, the values of these parameters is not specified in the Registry, rather these parameters are listed as an aid to the measurement system implementor or user (they must be left as variables, and supplied on execution).

Where metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Run-Time Parameters.

The Data Format of each Run-time Parameter SHALL be specified in this column, to simplify the control and implementation of measurement devices.

Examples of Run-time Parameters include IP addresses, measurement point designations, start times and end times for measurement, and other information essential to the method of measurement.

#### 4.4. Comments and Remarks

Besides providing additional details which do not appear in other categories, this open Category (single column) allows for unforeseen issues to be addressed by simply updating this Informational entry.

## 5. Example Generalized Passive Octet Count Entry

tbd

This section is Informational.

This section gives an example registry entry for a generalized the passive metric octetDeltaCount described in [RFC5102].

### 5.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

#### 5.1.1. Element Identifier

An integer having enough digits to uniquely identify each entry in the Registry.

TBD by IANA.

#### 5.1.2. Metric Name

A metric naming convention is TBD.

One possibility based on the proposal in [I-D.manyfolks-ippm-metric-registry]:

Pas\_IP-Octet-Delta-General

#### 5.1.3. Status

Current

#### 5.1.4. Requester

TBD

#### 5.1.5. Revision

0

#### 5.1.6. Revision Date

TBD

#### 5.1.7. Metric Description

A delta count of the number of octets observed.

#### 5.1.8. Reference Specification(s)

octetDeltaCount described in section 5.10.1 of [RFC5102]

#### 5.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

##### 5.2.1. Reference Definition

octetDeltaCount described in section 5.10.1 of [RFC5102]

##### 5.2.2. Fixed Parameters

As this is the generalised version of the IP delta count metric, there are no fixed parameters.

#### 5.3. Method of Measurement

##### 5.3.1. Reference Implementation

For <metric>.

<section reference>

##### 5.3.2. Traffic Filter Criteria

This measurement only covers IP packets and the IP payload (including the IP header) of these packets. Non-IP packets (BPDUs, ISIS) will not be accounted. Layer 2 overhead (Ethernet headers, MPLS, QinQ, etc.) will also not be represented in the measurement.

##### 5.3.3. Measurement Timing

This is a continuous measurement of the IP octets seen in the traffic selection scope (run-time parameter).

The measurement interval is a run time parameter.

There is no sampling.

#### 5.3.4. Output Type(s) and Data Format

It is possible that multiple observation intervals are reported in a single report. In such a case concatenation of the interval reports (deltaOctetCount, start-time, end-time) is allowed.

The delta octet count metric reports a observation start time and end time.

- o Value: observation-start-time and observation-end-time
- o Data Format: 64-bit NTP Time-stamp Format
- o Reference: section 6 of [RFC5905]

#### 5.3.5. Metric Units

The measured results are expressed in octets with a data format of unsigned64 as described in [RFC5102]

#### 5.3.6. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

- o samplingTimeInterval, length of time a single report covers. unsigned32 microseconds [RFC5477]
- o observationInterface, ifindex of interface to monitor. -1 represents all interfaces. -2 representings WAN facing and -3 represnets LAN facing. unsigned32.
- o observation direction, unsigned8 where 0 represents incoming traffic on interface, 1 outgoing and 2 represents both incoming and outgoing.

#### 5.4. Comments and Remarks

Additional (Informational) details for this entry

#### 6. Example 5min Passive Egress Octet Count Entry on WAN Interface

tbd

This section is Informational.

This section gives an example registry entry for accounting of outgoing WAN IP traffic the passive metric in terms of octetDeltaCount, as described in [RFC5102].

#### 6.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

##### 6.1.1. Element Identifier

An integer having enough digits to uniquely identify each entry in the Registry.

TBD by IANA.

##### 6.1.2. Metric Name

A metric naming convention is TBD.

One possibility based on the proposal in [I-D.manyfolks-ippm-metric-registry]:

Pas\_IP-Octet-Delta-WAN-egress

##### 6.1.3. Status

Current

##### 6.1.4. Requester

TBD

##### 6.1.5. Revision

0

##### 6.1.6. Revision Date

TBD

##### 6.1.7. Metric Description

A delta count of the number of octets observed outgoing on WAN interface.

#### 6.1.8. Reference Specification(s)

octetDeltaCount described in section 5.10.1 of [RFC5102]

#### 6.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

##### 6.2.1. Reference Definition

octetDeltaCount described in section 5.10.1 of [RFC5102]

##### 6.2.2. Fixed Parameters

As this is a specific version of Pas\_IP-Octet-Delta-General that performs metering of all outgoing WAN traffic.

- o samplingTimeInterval= 300000000, length of time a single report covers. unsigned32 microseconds [RFC5477]
- o observationInterface= -2, ifindex of interface to monitor. -1 represents all interfaces. -2 representings WAN facing and -3 represnets LAN facing. unsigned32.
- o observation direction= 1, unsigned8 where 0 represents incoming traffic on interface, 1 outgoing and 2 represents both incoming and outgoing.

#### 6.3. Method of Measurement

##### 6.3.1. Reference Implementation

For <metric>.

<section reference>

##### 6.3.2. Traffic Filter Criteria

This measurement only covers IP packets observed in the WAN outgoing direction. The bytes counted are the IP payload (including the IP header) of these packets. Non-IP packets (BPDUs, ISIS) will not be accounted. Layer 2 overhead (Ethernet headers, MPLS, QinQ, etc.) will also not be represented in the measurement.

#### 6.3.3. Measurement Timing

This is a continuous measurement of the IP octets seen in the traffic selection scope (run-time parameter), each of a 5 minute duration.

There is no sampling.

#### 6.3.4. Output Type(s) and Data Format

It is possible that multiple observation intervals are reported in a single report. In such a case concatenation of the interval reports (deltaOctetCount, start-time, end-time) is allowed.

The delta octet count metric reports a observation start time and end time.

- o Value: observation-start-time and observation-end-time
- o Data Format: 64-bit NTP Time-stamp Format
- o Reference: section 6 of [RFC5905]

#### 6.3.5. Metric Units

The measured results are expressed in octets with a data format of unsigned64 as described in [RFC5102]

#### 6.3.6. Run-time Parameters and Data Format

There are no run-time parameters for this registry entry.

#### 6.4. Comments and Remarks

Additional (Informational) details for this entry

#### 7. Example Passive RTP Lost Packet Count

tbd

#### 8. Example BLANK Registry Entry

This section is Informational. (?)

This section gives an example registry entry for the <type of metric and specification reference> .



## 8.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

### 8.1.1. Element Identifier

An integer having enough digits to uniquely identify each entry in the Registry.

### 8.1.2. Metric Name

A metric naming convention is TBD.

### 8.1.3. Status

Current

### 8.1.4. Requester

TBD

### 8.1.5. Revision

0

### 8.1.6. Revision Date

TBD

### 8.1.7. Metric Description

A metric Description is TBD.

### 8.1.8. Reference Specification(s)

Section YY, RFCXXXX

## 8.2. Metric Definition

### 8.2.1. Reference Definition

< possible section reference >

### 8.2.2. Fixed Parameters

Fixed Parameters are input factors that must be determined and embedded in the measurement system for use when needed. The values of these parameters is specified in the Registry.

<list fixed parameters>

### 8.3. Method of Measurement

#### 8.3.1. Reference Implementation

For <metric>.

<section reference>

#### 8.3.2. Traffic Filter Criteria

<list filter criteria limitations and allowances >

#### 8.3.3. Measurement Timing

< list timing requirements and limitations >

#### 8.3.4. Output Type(s) and Data Format

The output types define the type of result that the metric produces.

- o Value:
- o Data Format: (There may be some precedent to follow here, but otherwise use 64-bit NTP Time-stamp Format, see section 6 of [RFC5905]).
- o Reference: <section reference>

#### 8.3.5. Metric Units

The measured results are expressed in <units>.

<section reference>.

#### 8.3.6. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

<list of run-time parameters>

<reference(s)>.

#### 8.4. Comments and Remarks

Additional (Informational) details for this entry

#### 9. Security Considerations

This registry has no known implications on Internet Security.

#### 10. IANA Considerations

IANA is requested to create The Passive Performance Metric Sub-registry within the Performance Metric Registry defined in [I-D.manyfolks-ippm-metric-registry]. The Sub-registry will contain the following categories and (bullet) columns, (as defined in section 3 above):

Common Registry Indexes and Info

- o Identifier
- o Name
- o Status
- o Requester
- o Revision
- o Revision Date
- o Description
- o Reference Specification(s)

Metric Definition

- o Reference Definition
- o Fixed Parameters

Method of Measurement

- o Reference Implementation

- o Traffic Filter Criteria
- o Measurement Timing
- o Output Type(s) and Data format
- o Metric Units
- o Run-time Parameters

Comments and Remarks

## 11. Acknowledgements

The authors thank the prior work of Al Morton, Marcelo Bagnulo and Phil Eardley in "draft-mornuley-ippm-registry-active" which was used both as a template for this document and source of text.

## 12. References

### 12.1. Normative References

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

### 12.2. Informative References

- [BMWG] IETF, , "Benchmarking Methodology (BMWG) Working Group, <http://datatracker.ietf.org/wg/bmwg/charter/>", .
- [I-D.ietf-lmap-framework] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-03 (work in progress), January 2014.
- [I-D.manyfolks-ippm-metric-registry] Bagnulo, M., Claise, B., Eardley, P., and A. Morton, "Registry for Performance Metrics", draft-manyfolks-ippm-metric-registry-00 (work in progress), February 2014.
- [I-D.mornuley-ippm-registry-active] Morton, A., Bagnulo, M., and P. Eardley, "Active Performance Metric Sub-Registry", draft-mornuley-ippm-registry-active-00 (work in progress), February 2014.
- [IPFIX] IETF, , "IP Flow Information eXport (IPFIX) Working Group, <http://datatracker.ietf.org/wg/ipfix/charter/>", .

- [PMOL] IETF, , "IP Performance Metrics for Other Layers (PMOL) Working Group,  
<http://datatracker.ietf.org/wg/pmol/charter/>", .
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC6035] Pendleton, A., Clark, A., Johnston, A., and H. Sinnreich, "Session Initiation Protocol Event Package for Voice Quality Reporting", RFC 6035, November 2010.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.
- [XRBLOCK] IETF, , "Metric Blocks for use with RTCP's Extended Report Framework (XRBLOCK),  
<http://datatracker.ietf.org/wg/xrblock/charter/>", .

Authors' Addresses

Aamer Akhter  
Cisco Systems, Inc.  
7025 Kit Creek Road  
RTP, NC 27709  
USA

Email: aakhter@cisco.com

Benoit Claise  
Cisco Systems, Inc.  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

Phone: +32 2 704 5622  
Email: bclaise@cisco.com

IPPM  
Internet-Draft  
Intended status: Informational  
Expires: August 16, 2014

L. Deng  
Z. Cao  
China Mobile  
February 12, 2014

Problem Statement for IP measurement in mobile networks  
draft-deng-ippm-wireless-01.txt

Abstract

This document analyzes the potential problems of applying existing IP-based performance measurement methods to wireless accessing environments. It suggests that a more flexible passive measuring framework and performance metrics, such as congestion ratio are needed.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 16, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Motivation . . . . .	3
2.1. Dynamic Load Balancing . . . . .	3
2.2. Radio Congestion Detection . . . . .	4
2.3. Accurate Troubleshooting . . . . .	5
2.4. Summary . . . . .	6
3. Further Considerations . . . . .	7
3.1. Congestion ratio metric . . . . .	7
3.2. Multi-hop Measurement Framework . . . . .	7
4. Security Considerations . . . . .	7
5. IANA Considerations . . . . .	7
6. References . . . . .	8
6.1. Normative References . . . . .	8
6.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

It is well-accepted that mobile Internet usage is going to increase fast in the coming years and replace the traditional voice service to be the dominant revenue source for mobile operators. In the meantime, fast evolving network and terminal technologies and changing service trend (e.g. social networking, video on demand, online reading, etc.) results in higher user service requirement. Therefore, as the basic infrastructure service provider, operators are deemed responsible for mobile Internet end-to-end performance, for subscribers want to get what they want, which gives rise to a basic yet important question: how does network service provider manage end-to-end service quality? In particular, there are two goals for operator's quality management initiative:

- o to make sure and validate the QoS metrics of specific IP flows against the values pre-defined by the service SLA(Service Level Agreement) from the user/service provider's point of view; and
- o to make sure and validate the sanity of network devices/links.

In this draft, we present three usecases and the potential problems of applying existing IP-based performance measurement methods to wireless accessing environments, where resource pooling and dynamic load balancing techniques are employed to accommodate explosively increasing data traffic, and suggest requirements for more robust passive measuring methods and performance metrics for such environment.



## 2. Motivation

### 2.1. Dynamic Load Balancing

Pooling technology has been introduced to the user plane in the packet switched domain of operator's core network for cellular subscribers since 3GPP Release 5 (3GPP TS23.236). With pooling, the traffic path from user equipments to the Internet via core network is not static, but rather dynamically assigned to a proper instance of an device pool, according to load balancing policies. The assignment is dynamically made at the time of user equipment's attachment establishment with the cellular core network, and would remain unchanged unless the mobile terminal detaches from the network or moves outside the base-stations' coverage subordinating to the specific core network's device pool.

As shown by Figure 1, potential device pools along the path all the way from the user terminal via the packet switching domain of the mobile network core to a third party service provider over the Internet. Examples of network devices that can be poolized include SGSN(Serving GPRS Support Node) and GGSN(Gateway GPRS Supporting Node). Moreover, the service provider could also implement load balancing on the server's side either via server-pooling within a data center or via (third party) CDN nodes.

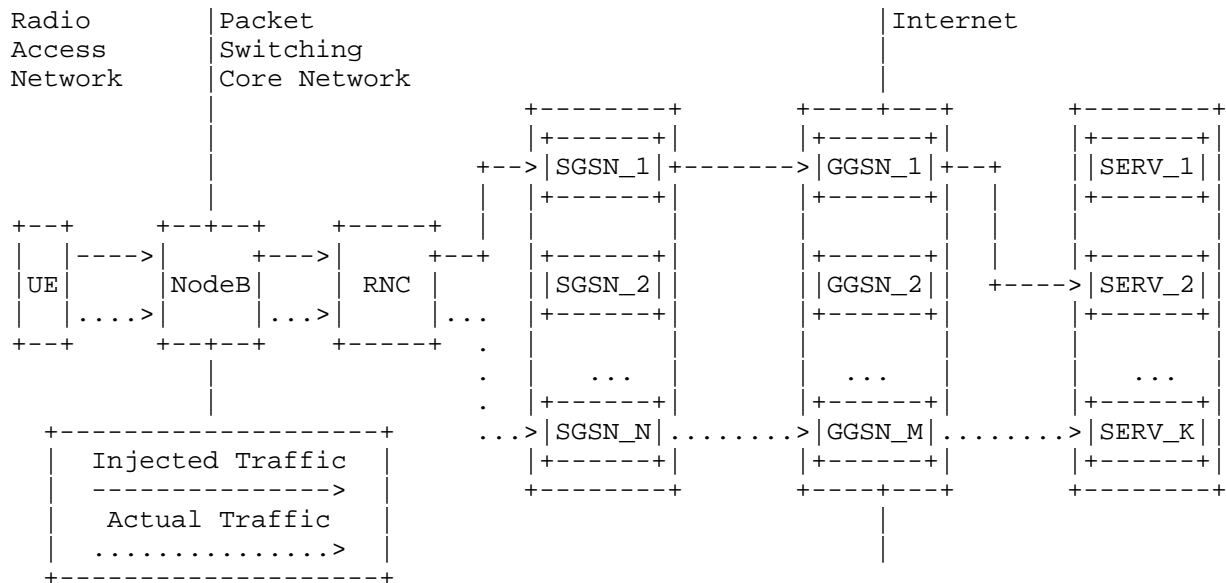


Figure 1: Active Measuring Traffic versus Actual Traffic in case of Device Pooling

Hence, under such environments, if active performance measurement methods[RFC4656][RFC5357] are employed, the injected bogus data traffic may traverse along a different path to the one used by the targeted traffic or even interfere with them due to the subtle nature of wireless-involved links (as explained in the next subsection).

## 2.2. Radio Congestion Detection

Mobile Internet usage is going to increase fast in the coming years due to the following facts: on one hand, as a result of pervasively deployed and fast maturing 3G/4G cellular technologies combined with smartphone's dominance in mobile handset's market, Internet data traffic via mobile operator's packet switched core network manifests to be an increasingly important contributor to the operator's revenue. On the other hand, wireless technologies (such as Wi-Fi through APs or cellular networks through small cells) are more and more accepted by the end users, either at home, in the office or in a public place, to be carrying the "last mile" to various portable personal computing devices.

There are two common features of the above two scenarios:

- o the combination of both wireless and wired links along the end-to-end traffic path, and
- o almost all the time, the wireless "last mile" would be the bottleneck of end-to-end service quality.

To make more efficient use of relatively more scarce radio resources, it is important for the core network to understand the congestion status of both wireless and wired links along the traffic path, and make proper management of data traffic through cell reselection or load balancing via pooling.

However, the wireless link's throughput is consistently subject to other interfering factors (e.g. distance to the nearest base station, terminal's radio signal strength, random interference, shadowing of buildings, multipath fading, etc.), which should be properly filtered out before handing over to the network management, as they are rooted in terminal mobility and outside the realm of mobile accessing network.

In other words, there is considerable gap between IP measurement results to the performance evaluation and fault detection requirements in mobile-involved environment, if we directly employ existing passive performance measurement methods[I-D.draft-chen-ippm-coloring-based-ipfpm-framework].

### 2.3. Accurate Troubleshooting

As shown in Figure 2, it is quite common that there are path partitions (belonging to different operation and management departments) along the local data path from the UE to the Internet within an mobile operator's local network. For large operators, employing layered network operation and management architecture based on geographic partitions, there may be a further more subpath partitioning between local IP backhaul (managed by state sub-ordinaries) and national IP backhaul (managed by header quarters). Moreover, for roaming cases under home-routed mode (meaning all the traffic from a roaming UE would first traverse from the visited ISP and potentially another Internet operator before getting back to homing ISP network).

Take the example of a mobile subscriber getting access from a 3GPP network for example, besides a local mobile network operator, intermediary ISPs may exist between its traffic before it reaches the Internet. Moreover, within the local operator's network, radio access network (RAN), RAN backhaul and local core network could actually be constructed and managed by stuff from different departments, for they mainly come from different technical background.

In such complex situations, it can become frustrating to respond quickly to a simple UoE complaint, due to the exponentially exploded complexity to accurately locate the potential faults/congestion in a transient wireless-involved end2end data path.

On the other hand, tunnels, including GRE [RFC2784], GTP [TS29.060], IP-in-IP [RFC2003] or IPSec [RFC4301] etc, are widely deployed in 3GPP networks. And in 3GPP network tunnels are used to carry end user flows within the backhaul network. Tunnels brings another complexity in realizing effective troubleshooting using end2end passive methods.

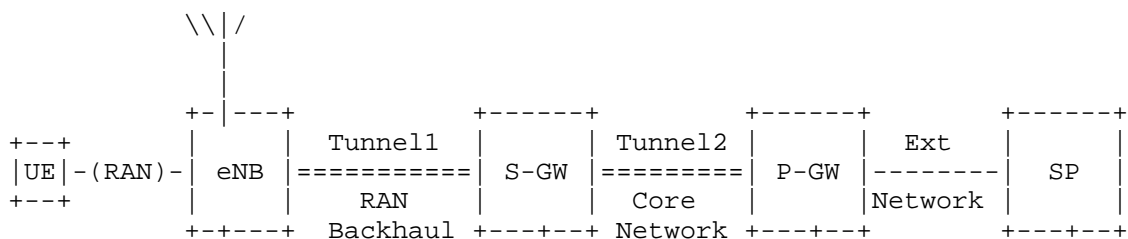


Figure 2: Example of path partition in 3GPP network

In other words, a flexible passive measurement framework, capable of dynamic troubleshooting for partitioned data link, even in case of tunnels or autonomous entities is highly valuable. However, neither current active measurement framework as used by OWAMP[RFC4656]/TWAMP[RFC5357], nor the passive framework proposed in [I-D.draft-chen-ippm-coloring-based-ipfpm-framework] could fit in such case.

#### 2.4. Summary

In summary, for mobile-ended data paths, we believe there is need for

- o viable passive measurement framework for active measurements inject extra traffic, which may traverse along a different path to the one used by the targeted traffic or even interfere with them.
- o robust metric against transient wireless conditions, as there is considerable gap between existing IP measurement metrics (e.g. delay, jitter, throughput etc.), which are subject to change caused by external environmental factors and of little use to operator's traffic management from the network side.

- o flexible and trustworthy measurement mechanisms for accurate performance monitoring and troubleshooting from multi-hop data link across operation boundaries.

### 3. Further Considerations

#### 3.1. Congestion ratio metric

ECN signal for congestion measurement are signalled at IP header by intermediary devices before actual congestion occurs, which is expected to be an effective indicator to potential QoE degradation, irrespective to traffic pattern/wireless conditions.

[I-D.draft-hedin-ippm-type-p-monitor] proposes to echo ECN-flags into TWAMP-test feedback for active measurement. While, packet-level echoing is not viable in passive framework, it is also suspected that more meaningful aggregated information (such as congestion extent, defined as the ratio of marked packets versus all packets from a given IP flow) would be preferred.

#### 3.2. Multi-hop Measurement Framework

In current active measurement framework, there is only two entities on the data path, the sender and the reflector. Hence it is not straightforward how to apply this framework to an integral multi-hop passive measurement case.

On the other hand, the centralized multi-hop passive framework proposed in [I-D.draft-chen-ippm-coloring-based-ipfpm-framework] could encounter problems when there is no prior knowledge about or control over different partitions along the overall data path. In other words, path discovery mechanism is needed to identify potential measurement nodes along the way during/before the actual passive measurement.

### 4. Security Considerations

If measurement nodes from different operational domains are used, proper device authentication and report authenticity protection mechanisms should also be considered in a complete interworking-capable solution.

### 5. IANA Considerations

None.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

### 6.2. Informative References

- [I-D.draft-chen-ippm-coloring-based-ipfpm-framework]  
Chen, M., Liu, H., Yin, Y., Papneja, R., Abhyankar, S.,  
and G. Deng, "Coloring based IP Flow Performance  
Measurement Framework", draft-chen-ippm-coloring-based-  
ipfpm-framework-01 (work in progress), October 2013.
- [I-D.draft-hedin-ippm-type-p-monitor]  
Hedin, J., Mirsky, G., and S. Baillargeon, "Differentiated  
Service Code Point and Explicit Congestion Notification  
Monitoring in Two-Way Active Measurement Protocol  
(TWAMP)", draft-hedin-ippm-type-p-monitor-02 (work in  
progress), October 2013.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.  
Zekauskas, "A One-way Active Measurement Protocol  
(OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.  
Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",  
RFC 5357, October 2008.

## Authors' Addresses

Lingli Deng  
China Mobile

Email: denglingli@chinamobile.com

Zhen Cao  
China Mobile

Email: caozhen@chinamobile.com

INTERNET-DRAFT

N. Elkins  
B. Jouris  
Inside Products  
K. Haining  
U. S. Bank  
M. Ackermann  
BCBS Michigan  
January 30, 2014

Intended Status: Proposed Standard  
Expires: July 2014

IPPM Considerations for the IPv6 PDM Extension Header  
draft-elkins-ippm-pdm-metrics-04

Table of Contents

1	Background . . . . .	5
1.1	Terminology . . . . .	5
1.2	Why End-to-end Response Time is Needed . . . . .	5
1.3	Trending of Response Time Data . . . . .	6
1.4	What to measure? . . . . .	6
1.5	TCP Timestamp not enough . . . . .	6
1.6	Inadequacy of Current Instrumentation Technology . . . . .	7
1.6.1	Synthetic transactions . . . . .	7
1.6.2	PING . . . . .	7
1.6.3	Estimates of Network Time . . . . .	8
1.6.4	Server / Client Agents . . . . .	8
2	Solution Parameters . . . . .	9
2.1	Rationale for proposed solution . . . . .	9
2.2	Merits of timestamp / delta in PDM . . . . .	9
2.3	What kind of timestamp? . . . . .	10
2	Why Packet Sequence Number . . . . .	10
2.1	IPv4 IPID : DeFacto Sequence Number . . . . .	11
2.1.1	Description of IPID in IPv4 . . . . .	11
2.1.2	DeFacto Use of IPID . . . . .	11
2.1.3	Merits of DeFacto Usage . . . . .	12
2.1.4	Use Cases of IPv4 IPID in Diagnostics . . . . .	12
2.2	TCP sequence number is not enough . . . . .	14
2.3	Inadequacy of current measurement techniques . . . . .	14
2.3.1	SNMP / CMIP Counters . . . . .	15
2.3.2	Router / Firewall Logs . . . . .	15
2.3.3	Netflow . . . . .	15
2.3.4	Access to Intermediate Devices . . . . .	15
2.3.4	Modifications to an Operational Production Network . . . . .	16
3	Solution Parameters . . . . .	16
3.1	Packet Trace Meets Criteria . . . . .	17
3.1.1	Limitations of Packet Capture . . . . .	17
3.1.2	Problem Scenario 1 . . . . .	17
3.1.2	Problem Scenario 2 . . . . .	17

4	Rationale for Proposed Solution (PDM)	18
5	Performance and Diagnostic Metrics Destination Option Layout	18
5.1	Destination Options Header	18
5.2	PDM Types	19
5.3	Performance and Diagnostic Metrics Destination Option (Type 1)	19
5.4	Performance and Diagnostic Metrics Destination Option (Type 2)	21
6	Use of the PDM	24
6.1	Packet Identification Data	24
6.2	Data in the PDM Destination Option Headers	24
7	Metrics Derived from the PDM Destination Options	25
8	Base Derived Metrics	25
8.1	One-Way Delay	25
8.2	Round-Trip Delay	25
8.3	Server Delay	26
9	Sample Implementation Flow (PDM Type 1)	26
9.1	Step 1 (PDM Type 1)	26
9.2	Step 2 (PDM Type 1)	27
9.3	Step 3 (PDM Type 1)	28
9.4	Step 4 (PDM Type 1)	29
9.5	Step 5 (PDM Type 1)	30
10	Sample Implementation Flow (PDM 2)	30
10.1	Step 1 (PDM Type 2)	30
10.2	Step 2 (PDM Type 2)	31
10.3	Step 3 (PDM Type 2)	32
10.4	Step 4 (PDM Type 2)	33
10.5	Step 5 (PDM Type 2)	34
11	Derived Metrics : Advanced	34
11.1	Advanced Derived Metrics : Triage	34
11.2	Advanced Derived Metrics : Network Diagnostics	35
11.2.1	Retransmit Duplication (RD)	35
11.2.2	ACK Lag (AL)	36
11.2.3	Third-party Connection Reset (TPCR)	36
11.2.4	Potential Hang (PH)	37
11.3	Advanced Metrics : Session Classification	37
12	Use Cases	37
13	Security Considerations	38
14	IANA Considerations	38
15	References	38
15.1	Normative References	38
15.2	Informative References	39
16	Acknowledgments	39
	Authors' Addresses	39

Abstract



To diagnose performance and connectivity problems, metrics on real (non-synthetic) transmission are critical for timely end-to-end problem resolution. Such diagnostics may be real-time or after the fact, but must not impact an operational production network. These metrics are defined in the IPv6 Performance and Diagnostic Metrics Destination Option (PDM). The base metrics are: packet sequence number and packet timestamp. Other metrics may be derived from these for use in diagnostics. This document specifies such metrics, their calculation, and usage.

#### Status of this Memo

This Internet-Draft is submitted to IETF 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.

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/lid-abstracts.html>

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

## Copyright and License Notice

Copyright (c) 2014 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.

## 1 Background

To diagnose performance and connectivity problems, metrics on real (non-synthetic) transmission are critical for timely end-to-end problem resolution. Such diagnostics may be real-time or after the fact, but must not impact an operational production network. The base metrics are: packet sequence number and packet timestamp. Metrics derived from these will be described separately. This document starts with the background and rationale for the requirement for end-to-end response time and packet sequence number(s).

Current methods are inadequate for these purposes because they assume unreasonable access to intermediate devices, are cost prohibitive, require infeasible changes to a running production network, or do not provide timely data. The IPv6 Performance and Diagnostic Metrics destination option PDM) provides a solution to these problems.

### 1.1 Terminology

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

### 1.2 Why End-to-end Response Time is Needed

The timestamps or delta values in the PDM traveling along with the packet will be used to calculate end-to-end response time, without requiring agents in devices along the path. In many networks, end-to-end response times are a critical component of Service Levels Agreements (SLAs).

End-to-end response is what the user of a network system actually experiences. When the end user is an individual, he is generally indifferent to what is happening along the network; what he really cares about is how long it takes to get a response back. But this is not just a matter of individuals' personal convenience. In many cases, rapid response is critical to the business being conducted.

When the end user is a device (e.g. with the Internet of Things), what matters is the speed with which requested data can be transferred -- specifically, whether the requested data can be transferred in time to accomplish the desired actions. This can be important when the relevant external conditions are subject to rapid change.

Response time and consistency are not just "nice to have". On many networks, the impact can be financial hardship or endanger human life. In some cities, the emergency police contact system operates

over IP, law enforcement uses TCP/IP networks, transactions on our stock exchanges are settled using IP networks. The critical nature of such activities to our daily lives and financial well-being demand a solution. Section 1.5 will detail the current state of end-to-end response time monitoring today.

### 1.3 Trending of Response Time Data

In addition to the need for tracking current service, end-to-end response time is valuable for capacity planning. By tracking response times, and identifying trends, it becomes possible to determine when network capacity is being approached. This allows additional capacity to be obtained before service levels fall below requirements. Without that kind of tracking, the only option is to wait until there is a problem, and then scramble to get additional capacity on an emergency (and probably high cost) basis.

### 1.4 What to measure?

End to end response time can be broken down into 3 parts:

- Network delay - Application (or server) delay- Client delay

Network delay may be one-way delay [RFC2679] or round-trip delay [RFC2681].

Additionally, network delay may include multiple hops. Application and server delay include operating system by stack time. By and large, the three timings are 'good enough' measurements to allow rapid triage into the failing component.

Ways are available (provided by operating systems) to measure Application and Client times. Network time can also be measured in isolation via some of the measurement techniques described in section 1.5. The most difficult portion is to integrate network time with the server or application times. Products exist to do this but are available at an exorbitant cost, require agents, and will likely become more prohibitive as the speed of networks grow and as the world becomes more connected via mobile devices.

Measuring network time needs to occur at the end-points of the transactions being measured. The time needs to be available, regardless of the upper layer protocol being used by the transaction. That is, it cannot be for just TCP packets.

### 1.5 TCP Timestamp not enough

Some suggest that the TCP Timestamp option might be sufficient to

calculate end-to-end response time.

The TCP Timestamp Option is defined in RFC1323 [RFC1323]. The reason for the TCP Timestamp option is to be able to discard packets when the TCP sequence number wraps. (PAWS)

The problems with the TCP Timestamp option are:

1. Not everyone turns this on.
2. It is only available for TCP applications
3. No indication of date in long-running connections. (That is connections which last longer than one day)
4. The granularity of the timestamp is at best at millisecond level.

In the future, as speeds of devices and networks grow and network types proliferate, TCP timestamp values, both in terms of granularity and date specification, will become more and more inadequate. Even today, on many networks, the timings are at microsecond level not millisecond. New networks called Delay Tolerant Networks may have connection times which are very large indeed - hours or even days.

## 1.6 Inadequacy of Current Instrumentation Technology

The current technology includes:

1. Synthetic transactions
2. Pings
3. Estimates of network time
4. Server / Client Agents

Let us discuss each of these in detail.

### 1.6.1 Synthetic transactions

Synthetic transactions, also known as active measurement, can be extremely useful. However, in a dynamic network, the routes taken by the packet or the current load on the application may not be the same for the real transaction as when the active test was performed. For example, if you time how long it takes for me to drive to work at 2:00am in the morning, that may not be the same as how long it takes me to drive to work during rush hour at 8:00am in the morning. So, it is important to have embedded measurement in the actual packet.

### 1.6.2 PING

An ICMP ping measures network time. First, you can PING the remote device. Then you assume that the time it takes to get a response to a PING is the same as the time that a transaction would take to

traverse the network. However, QoS rules, firewalls, etc. may mean that PING, (and other synthetic transactions) may not be subject to the same conditions. PINGs, though extremely useful, also measure only network delays. Server delays must also be provided.

#### 1.6.3 Estimates of Network Time

If a packet trace is done, it is possible to look at the time between when a response was seen to be sent at the packet capture device and when the ACK for the response comes back.

If you assume that the ACK took the same amount of time as the original query, you have the network time. Unfortunately, the time for the ACK may not be the same as the time for a much larger query transaction to traverse the network.

The biggest problem with this method is that of TCP delayed acknowledgements. If the client is doing delayed ACKs, then the ACK will be held until the next request is ready to go out. In this case, the time to receive the ACK has no correlation with network time.

#### 1.6.4 Server / Client Agents

There are also products which claim that they can determine end-to-end response times, integrating server and network times - and indeed they can do so. But they require agents which must be placed at each point which is to be monitored. That is, it is necessary to add those agents EVERYWHERE around the network, at a very high cost - both in terms of manpower, knowledge and costs. These kind of products can be purchased by only the richest 1% of the corporations. As the speed of networks grow, and as the world becomes more connected via mobile devices, such products will only become more expensive. If, indeed, their technology can keep up.

There are many situations where agents cannot be deployed. Many situations which demand a lightweight, cost effective solution. You may think of an ISP with many customers. If the customer complains of poor response time, it is much more cost-effective for the ISP to simply take a packet trace with embedded diagnostics than to instrument the entire customer network.

TCP/IP networks, including the Internet, are used throughout the world. If there is not a scalable and affordable way to measure performance bottlenecks and failures, the growth of these networks will suffer and indeed may reach a plateau where further growth becomes impossible.

## 2 Solution Parameters

What is needed is:

- 1) A method to identify and/or track the behavior of a connection without assuming access to the transport devices.
- 2) A method to observe a connection in flight without introducing agents.
- 3) a method to observe arbitrary flows at multiple points within a network and correlate the results of those observations in a consistent manner.
- 4) A method to signal and correlate transport issues to application end-to-end behavior.
- 5) A method which does not require changes to a production network in real time.
- 6) Adequate granularity in the measurement technique to provide the needed metrics.
- 7) A method that is scalable to very large networks.
- 8) A method that is affordable to all.

### 2.1 Rationale for proposed solution

The current IPv6 specification does not provide a timestamp nor similar field in the IPv6 main header or in any extension header. So, we propose the IPv6 Performance and Diagnostic Metrics destination option (PDM) [ELKPDM].

### 2.2 Merits of timestamp / delta in PDM

Advantages include:

1. Less overhead than other alternatives.
2. Real measure of actual transactions.
3. Less cost to provide solutions
4. More accurate and complete information.
5. Independence from transport layer protocols.
6. Ability to span organizational boundaries with consistent instrumentation

In other words, this is a solution to a long-standing problem. The PDM will provide a metric which will allow those responsible for

network support to determine what is happening in their network without expensive equipment (agents) at each device.

The PDM does not solve every response time issue for every situation. Network connections with multiple hops will still need more granular metrics, as will the differentiation between multiple components at each host. That is, TCP/IP stack time vs. applications time will still need to be broken out by client software. What the PDM does provide is the ability to do rapid triage. That is, to determine quickly if the problem is in the network or in the server or application.

### 2.3 What kind of timestamp?

Questions arise about exactly the kind of timestamp to use. Both the Network Time Protocol (NTP) [RFC5905] and Precision Time Protocol (PTP) [IEEE1588] are used to provide timing on TCP/IP networks.

NTP has evolved within the IETF structure while PTP has evolved within the Institute of Electrical and Electronics Engineers (IEEE) community. By and large, operating systems such as Windows, Linux, and IBM mainframe computers use NTP. These are the source and destination systems for packets. Intermediate nodes such as routers and switches may prefer PTP.

Since we are describing a new extension header for destination systems, the timestamp to be used will be in accordance with NTP. The document, draft-ackermann-ntp-pdm-ntp-usage [NTPPDM], discusses guidelines for implementing NTP for use with the PDM. The timestamp is only relevant for PDM type 1. PDM type 2 uses delta values and requires no time synchronization.

## 2 Why Packet Sequence Number

While performing network diagnostics of an end-to-end connection, it often becomes necessary to find the device along the network path creating problems. Diagnostic data may be collected at multiple places along the path (if possible), or at the source and destination. Then, the diagnostic data must be matched. Packet sequence number is critical in this matching process. The timestamp or even the IP addresses may be different at different devices. In IPv4 networks, the IPID field was used as a de facto sequence number.

This method of data collection along the path is of special use on large multi-tier networks to determine where packet loss or packet corruption is happening. Multi-tier networks are those which have multiple routers or switches on the path between the sender and the receiver.



## 2.1 IPv4 IPID : DeFacto Sequence Number

With IPv4 networks, on many stack implementations, but not all, the IPID field has the property of sequentiality. That is, the IP stack sending the packets sent them in numerical order. This was not a requirement for the field, but an implementation which turned out to be quite useful in diagnostics.

### 2.1.1 Description of IPID in IPv4

In IPv4, the 16 bit IP Identification (IPID) field is located at an offset of 4 bytes into the IPv4 header and is described in RFC0791 [RFC0791]. In IPv6, the IPID field is a 32-bit field contained in the Fragment Header defined by section 4.5 of RFC2460 [RFC2460]. Unfortunately, unless fragmentation is being done by the source node, the IPv6 packet will not contain this Fragment Header, and therefore will have no Identification field.

The intended purpose of the IPID field, in both IPv4 and IPv6, is to enable fragmentation and reassembly, and as currently specified is required to be unique within the maximum segment lifetime (MSL) on all datagrams. The MSL is often 2 minutes.

### 2.1.2 DeFacto Use of IPID

In a number of networks, the IPID field is used for more than fragmentation. During network diagnostics, packet traces may be taken at multiple places along the path, or at the source and destination. Then, packets can be matched by looking at the IPID.

The inclusion of the IPID makes it easier to identify flows belonging to a single node, even if that node might have a different IP address. For example, in the case of sessions going through a NAT or proxy server.

For its de-facto diagnostic mode usage, the IPID field needs to be available whether or not fragmentation occurs. It also needs to be unique in the context of the session, and across all the connections controlled by the stack. In IPv4, the IPID is in the main header, so it is available for all packets. As it is a 16-bit field, it wrapped during the course of the session and thus had some limitations.

Even with these limitations, the IPID has been valuable and useful in IPv4 for diagnostics and problem resolution. It is a practical solution that is 'good enough' in many instances. Not having it available in IPv6, may be a major detriment to new IPv6 deployments and contribute to protracted downtimes in existing IPv6 operations.

### 2.1.3 Merits of DeFacto Usage

As network technology evolves, the uses to which fields are put can change as well. De-facto use is powerful, and should not be lightly ignored. In fact, it is a testament to the power and pervasiveness of the protocol that users create new uses for the original technology.

For example, the use of the IPID goes beyond the vision of the original authors. This sort of thing has happened with numerous other technologies and protocols.

The implementation of the traceroute command sends ICMP echo packets with a varying TTL. This is a very useful for diagnostics yet departs from the original purpose of TTL.

Similarly, cell phones have evolved to be more than just a means of vocal communication, including Internet communications, photo-sharing, stock exchange transactions, etc. Indeed, the Internet itself has evolved, from a small network for researchers and the military to share files into the pervasive global information superhighway that it is today.

### 2.1.4 Use Cases of IPv4 IPID in Diagnostics

Use Case # 1 --- Large Insurance Company

- (estimated time saved by use of IPID: 7 hours)

Performance Tool produces extraneous packets

- Issue was whether a performance tool was accurately replicating session flow during performance testing.
- Trace IPIDs showed more unique packets within same flow from performance tool compared to IE Browser.
- Having the clear IPID sequence numbers also showed where and why the extra packets were being generated.
- Solution: Problem rectified in subsequent version of performance tool.
- Without IPID, it was not clear if there was an issue at all.

Use Case #2 --- Large Bank

- (estimated time saved by use of IPID: 4 hours)

Batch transfer duration increases 12x

- A data transfer which formerly took 30 minutes to complete started taking 6-8 hours to complete.
- Was there packet loss? All the vendors said no.
- The other applications on the network did not report any problems.
- 4 trace points were used, and the IPIDs in the packets were compared.
- The comparison showed 7% packet loss.
- Solution: WAN hardware was replaced and problem fixed.
- Without IPID, no one would agree a problem existed

Use Case #3 --- Large Bank

- (estimated time saved by use of IPID: 6 hours)

Very slow interactive performance

- All network links looked good.
- Traces showed duplicated small packets (which can be OK).
- We saw that the IPID was the same in both packets but the TTL was always + 1.
- A network device was "splitting" only small packets over two interfaces.
- The small packets were control info, telling other side to slow down.
- It erroneously looked like network congestion.
- Solution: Network device replaced and good interactive performance restored.
- Without IPID, flows would have appeared OK.

Use Case #4 --- Large Government Agency

- (estimated time saved by use of IPID: 9 hours)

VPN drops

- Cell phone connections to law enforcement were being dropped. The connections were going through a VPN.
- All parties (both sides of VPN connection, application, etc.) said it was not their problem. The problem went on for weeks.
- Finally, we took a trace which showed packets with IPID and TTL that did not match others in the flow AT ALL coming from the router nearest the application server end of VPN.
- Solution: Provider for VPN for application server changed. Problem resolved.
- Without IPID, much harder to diagnose problem. Same case also happened with large corporation. Again, all parties saying not their fault until proven via packet trace.)

## 2.2 TCP sequence number is not enough

TCP Sequence number is defined in RFC0793 [RFC0793]. Some have proposed that this field will meet the needs of diagnostics for a packet sequence number. Indeed, the TCP Sequence Number along with the TCP Acknowledgment number can be used to calculate dropped packets, duplicate packets, out-of-order packets etc. That is, IF the packet flow itself reflects accurately what happened on the wire!

See Scenario 1 (Section 1.5.2) and Scenario 2 (Section 1.5.3) for what happens with packet trace capture in real networks.

The TCP Sequence Number is, obviously, available only for TCP and not other higher layer protocols.

## 2.3 Inadequacy of current measurement techniques

The question arises of whether current methods of instrumentation cannot be used without a change to the protocol. Current methods of measuring network data, other than packet traces, are inadequate because they assume unreasonable access to intermediate devices, are cost prohibitive, require infeasible changes to a running production network, or do not provide timely data. This section will discuss each of these in detail.

Current methods include both instrumentation and third party products. These include SNMP, CMIP, router logs, and firewall logs.

#### 2.3.1 SNMP / CMIP Counters

The traditional network performance counters measured by SNMP or CMIP do not provide information at the granularity desired on the behavior of application flows across the network. The problem is that such counters do not contain enough data to be able to provide a detailed and realistic view of the end-to-end behavior of a connection.

#### 2.3.2 Router / Firewall Logs

Router and firewall logs may provide some information for diagnostics. Routers and firewalls in a production network are generally set to do minimal logging and diagnostics to allow maximum efficiency and throughput. Such devices cannot be asked to collect detailed data for an operational problem, as this requires a change to a production network.

#### 2.3.3 Netflow

Netflow is instrumentation which is available from some middle devices. In production networks, such devices are generally set to do minimal logging and diagnostics to allow maximum efficiency and throughput.

It is often also not possible to start data collection in the middle of the day on a production network.

#### 2.3.4 Access to Intermediate Devices

The above current methods require access to the transport infrastructure - that is, the routers, switches or other intermediate devices. In some cases, this is possible; in others, the connections in question may cross a number of administrative entities (both in the transport and in the endpoints). When it is the enterprise at the endpoint which is interested in the diagnostics, the administrative entities who own the devices in the middle of the path have no stake in operational measurement at the enterprise or application level. They have no reason to provide the necessary data or to impact the basic transport with the instrumentation necessary to capture flow-oriented data as a continuous stream suitable for general consumption.

In other words, if you don't own the path end-to-end, you will not be able to get the data you need if you are required to get it from the devices in the middle. Not only that, the devices in the middle do

not have the instrumentation necessary to make it easy to do end-to-end diagnostics because they are not responsible for that and so do not want to burden their devices with doing those kind of functions.

Many networks may not own the path end-to-end. They may be working with a business partner's network or crossing the Internet.

#### 2.3.4 Modifications to an Operational Production Network

Even when the enterprise does own all the devices along the entire path, to get enough data to adequately resolve a problem means changing the device configuration to do detailed diagnostics. In a production network, devices are generally set to do minimal logging and diagnostics. This is to allow maximum efficiency and throughput. The more logging and diagnostics such devices do, the fewer resources they have for actually transmitting traffic across the network.

So, if devices are to be asked to collect more data for an operational problem, this requires a change to a production network. This is generally not possible as it destabilizes a critical network during business hours, thus potentially disrupting many customers. Making changes is usually a lengthy process requiring change control, testing on a test network, etc. On networks which are critical to the business function, changing configuration "in flight" is generally not an option.

### 3 Solution Parameters

What is needed is:

- 1) A method to identify and/or track the behavior of a connection without assuming access to the transport devices.
- 2) A method to observe a connection in flight without introducing agents at endpoints.
- 3) A method to observe arbitrary flows at multiple points within a network and correlate the results of those observations in a consistent manner.
- 4) A method to signal and correlate transport issues to application end-to-end behavior.
- 5) A method which does not require changes to a production network in real time.
- 6) Adequate granularity in the measurement technique to provide the needed metrics.

### 3.1 Packet Trace Meets Criteria

The only instrumentation which provides enough detail to diagnose end-to-end problems is a packet trace. Packet traces do not require changes to devices in production mode because in many networks, products are available to capture packets in passive mode. Such products continuously monitor network traffic. Often, they are used not for diagnostic reasons but for regulatory reasons. For example, there may be legal requirements to log all stock exchange transactions.

Products for packet tracing are available freely and can be used at a client host without disrupting major portions of the network.

#### 3.1.1 Limitations of Packet Capture

Even though packets are the only reliable way to provide data at the needed granularity, there are limitations with collecting packet traces in some situations. They are as follows:

##### 3.1.2 Problem Scenario 1

1. Packets are captured for analysis at places like large core switches. All packets are kept. Again, not necessarily for diagnostic reasons but for regulatory ones. For example, records of all stock trades may need to be kept for a certain number of years.
2. When there is a problem, an analyst extracts the needed information.
3. If the extract is done incorrectly, as often happens, or the packet capture itself is incorrect, then there may be false duplicate packets which can be quite misleading and can lead to wrong conclusions. Are these real TCP duplicates? Is there congestion on the subnet? Are these retransmissions? Situations have been seen where routers incorrectly send two packets instead of one - is this such a situation?
4. This is the type of problem that can be solved by having an IP packet sequence number.

##### 3.1.2 Problem Scenario 2

1. In this scenario, packets are captured for analysis at places like a middleware box. It may be because problems are suspected with the box itself or it is a central point of the suspected failure.
2. The box may not offer any way to tailor the packet capture. "You

will get what we give you, how we give it to you!" is their philosophy.

3. The packet capture incorrectly duplicates only packets going to certain nodes.

4. Again, there are false duplicate packets which can be misleading and can lead to wrong conclusions. Are these real TCP duplicates? Is there congestion on the subnet? Situations have been seen where routers incorrectly send two packets instead of one - is this such a situation?

#### 4 Rationale for Proposed Solution (PDM)

The current IPv6 specification does not provide a packet sequence number or similar field in the IPv6 main header. One option might be to force all IPv6 packets to contain a Fragment Header. In packets which are entire in and of themselves, the fragment ID would be zero - that is, an atomic fragment. Why was a new destination option header defined rather than recommending that Fragment Header be used?

Our reasoning was that the PDM destination option header would provide multiple benefits : the packet sequence number and the timings to calculate response time.

As defined in RFC2460 [RFC2460], destination options are carried by the IPv6 Destination Options extension header. Destination options include optional information that need be examined only by the IPv6 node given as the destination address in the IPv6 header, not by routers in between.

The PDM DOH will be carried by each packet in the network, if this is configured. That is, the PDM DOH is optional. If the user of the OS configures the PDM DOH to be used, then it will be carried in the packet.

The metrics in the PDM are for 'real' or passive data. That is, they are of the traffic actually traveling on the network.

#### 5 Performance and Diagnostic Metrics Destination Option Layout

##### 5.1 Destination Options Header

The IPv6 Destination Options Header is used to carry optional information that need be examined only by a packet's destination node(s). The Destination Options Header is identified by a Next Header value of 60 in the immediately preceding header and is defined in RFC2460 [RFC2460].



## 5.2 PDM Types

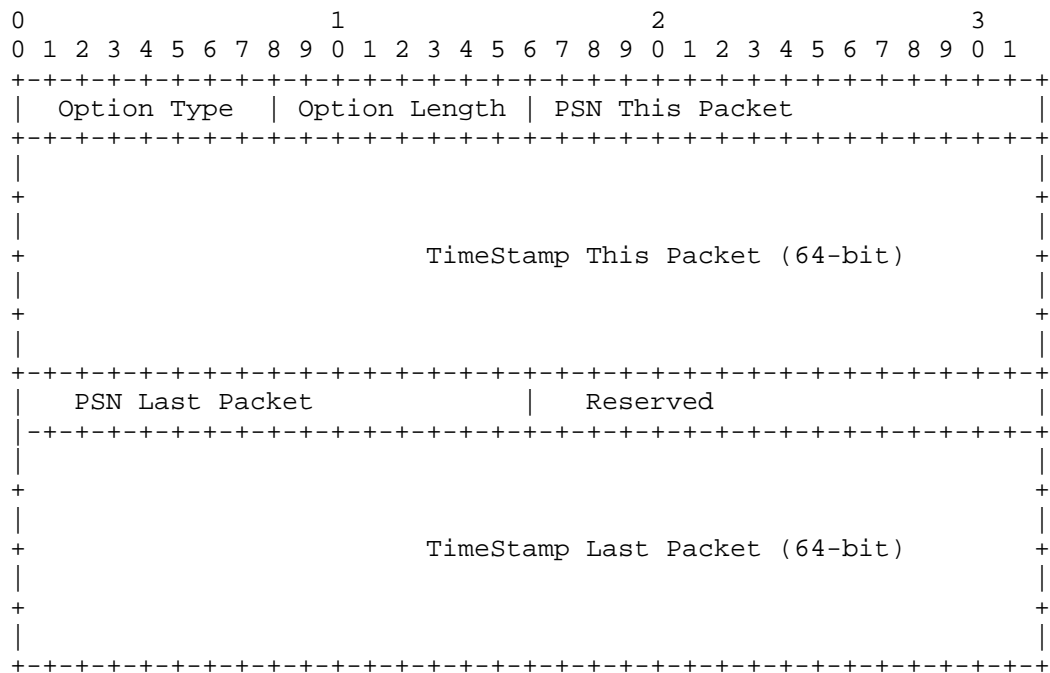
The IPv6 Performance and Diagnostic Metrics Destination Option (PDM) is an implementation of the Destination Options Header (Next Header value = 60). Two types of PDM are defined. PDM type 1 requires time synchronization. PDM type 2 does not require time synchronization.

PDM type 1 and PDM type 2 are mutually exclusive. That is, a 5-tuple can either both send PDM type 1 or both send PDM type 2.

## 5.3 Performance and Diagnostic Metrics Destination Option (Type 1)

PDM type 1 is used to facilitate diagnostics by including a packet sequence number and timestamp.

The PDM type 1 is encoded in type-length-value (TLV) format as follows:



Option Type

TBD = 0xXX (TBD) [To be assigned by IANA] [RFC2780]

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field MUST be set to 22.

#### Packet Sequence Number This Packet (PSNTP)

16-bit unsigned integer. This field will wrap. It is intended for human use.

Initialized at a random number and monotonically incremented for packet on the 5-tuple. The 5-tuple consists of the source and destination IP addresses, the source and destination ports, and the upper layer protocol (ex. TCP, ICMP, etc).

Operating systems MUST implement a separate packet sequence number counter per 5-tuple. Operating systems MUST NOT implement a single counter for all connections.

Note: This is consistent with the current implementation of the IPID field in IPv4 for many, but not all, stacks.

#### TimeStamp This Packet (TSTP)

A 64-bit unsigned integer field containing a timestamp that this packet was sent by the source node. The value indicates the number of seconds since January 1, 1970, 00:00 UTC, by using a fixed point format. In this format, the integer number of seconds is contained in the first 32 bits of the field, and the remaining 32 bits resolve to picoseconds.

This follows timestamp formats used in Network Time Protocol (NTP) [RFC5905] and SEND [RFC3971]. A discussion of how to implement NTP for use with PDM header type 1 is in draft-ackermann-ntp-pdm-ntp-usage-00 [NTPPDM].

Implementation note: This format is compatible with the usual representation of time under UNIX, although the number of bits available for the integer and fraction parts in different Unix implementations vary.

#### Packet Sequence Number Last Received (PSNLR)

16-bit unsigned integer. This is the PSN of the packet last received on the 5-tuple.

#### TimeStamp Last Received (TSLR)

A 64-bit unsigned integer field containing a timestamp. This is the timestamp of the packet last received on the 5-tuple. Format is the same as TSTP.

### 5.4 Performance and Diagnostic Metrics Destination Option (Type 2)

The second type of IPv6 Performance and Diagnostic Metrics Destination Option (PDM) is as follows. PDM type 1 and PDM type 2 are mutually exclusive. That is, a 5-tuple can either both send PDM type 1 or both send PDM type 2.

PDM type 2 contains the following fields:

```

PSNTP      : Packet Sequence Number This Packet
PSNLR      : Packet Sequence Number Last Received
DELTALR    : Delta Last Received
PSNLS      : Packet Sequence Number Last Sent
DELTALS    : Delta Last Sent

```

PDM destination option type 2 is encoded in type-length-value (TLV) format as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Option Type | Option Length | PSN This Packet |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| PSN Last Received | PSN Last Sent |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Delta Last Received | Delta Last Sent |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| TType |
+---+---+---+

```

#### Option Type

TBD = 0xXX (TBD) [To be assigned by IANA] [RFC2780]

#### Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field MUST be set to 22.

#### Packet Sequence Number This Packet (PSNTP)

16-bit unsigned integer. This field will wrap. It is intended for human use.

Initialized at a random number and monotonically incremented for packet on the 5-tuple. The 5-tuple consists of the source and destination IP addresses, the source and destination ports, and the upper layer protocol (ex. TCP, ICMP, etc).

Operating systems MUST implement a separate packet sequence number counter per 5-tuple. Operating systems MUST NOT implement a single counter for all connections.

Note: This is consistent with the current implementation of the IPID field in IPv4 for many, but not all, stacks.

#### Packet Sequence Number Last Received (PSNLR)

16-bit unsigned integer. This is the PSN of the packet last received on the 5-tuple.

#### Packet Sequence Number Last Sent (PSNLS)

16-bit unsigned integer. This is the PSN of the packet last sent on the 5-tuple.

#### Delta TimeStamp Type (TIMETYPE)

4-bit unsigned integer. This is the type of time contained in the delta fields below.

- 0 - unknown
- 1 - time is in units of nanoseconds
- 2 - time is in units microseconds
- 3 - time is in units of milliseconds
- 4 - time is in units of seconds
- 5 - time is in units of minutes
- 6 - time is in units of hours
- 7 - time is in units of days

The values 5 - 7 are relevant for Delay Tolerant Networks (DTN) which may operate with long delays between packets.

#### Delta Last Received (DELTALR)

A 16-bit unsigned integer field. This is server delay.

$\text{DELTALR} = \text{Send time packet 2} - \text{Receive time packet 1}$

The value is according to the scale in TIMETYPE.

Delta Last Sent (DELTALS)

A 16-bit unsigned integer field. This is round trip or end-to-end time.

$\text{Delta Last Sent} = \text{Receive time packet 2} - \text{Send time packet 1}$

The value is in according to the scale in TIMETYPE.

Option Type

The two highest-order bits of the Option Type field are encoded to indicate specific processing of the option; for the PDM destination option, these two bits MUST be set to 00. This indicates the following processing requirements:

00 - skip over this option and continue processing the header.

RFC2460 [RFC2460] defines other values for the Option Type field. These MUST NOT be used in the PDM. The other values are as follows:

01 - discard the packet.

10 - discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

11 - discard the packet and, only if the packet's Destination Address was not a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

In keeping with RFC2460 [RFC2460], the third-highest-order bit of the Option Type specifies whether or not the Option Data of that option can change en-route to the packet's final destination.

In the PDM, the value of the third-highest-order bit MUST be 0. The possible values are as follows:

0 - Option Data does not change en-route

1 - Option Data may change en-route

The three high-order bits described above are to be treated as part of the Option Type, not independent of the Option Type. That is, a particular option is identified by a full 8-bit Option Type, not just the low-order 5 bits of an Option Type.

## 6 Use of the PDM

### 6.1 Packet Identification Data

Each packet contains information about the sender and receiver. In IP protocol the identifying information is called a "5-tuple". The flows described below are for the set of packets flowing between A and B without consideration of any other packets sent to any other device from Host A or Host B.

The 5-tuple consists of:

SADDR : IP address of the sender  
SPORT : Port for sender  
DADDR : IP address of the destination  
DPORT : Port for destination  
PROTC : Protocol for upper layer (ex. TCP, UDP, ICMP, etc.)

### 6.2 Data in the PDM Destination Option Headers

The IPv6 Performance and Diagnostic Metrics Destination Option (PDM) is an implementation of the Destination Options Header (Next Header value = 60). Two types of PDM are defined. PDM type 1 requires time synchronization. PDM type 2 does not require time synchronization.

PDM type 1 and PDM type 2 are mutually exclusive. That is, a 5-tuple can either both send PDM type 1 or both send PDM type 2.

PDM type 1 contains the following fields:

PSNTP : Packet Sequence Number This Packet  
TSTP : Timestamp This Packet  
PSNLR : Packet Sequence Number Last Received  
TSLR : Timestamp Last Received

PDM type 2 contains the following fields:

PSNTP : Packet Sequence Number This Packet

PSNLR : Packet Sequence Number Last Received  
DELTALR : Delta Last Received  
PSNLS : Packet Sequence Number Last Sent  
DELTALS : Delta Last Sent

The metrics which may be derived from these fields will be discussed in the following sections.

## 7 Metrics Derived from the PDM Destination Options

A number of metrics may be derived from the data contained in the PDM. Some are relationships between two packets, others require analysis of multiple packets or multiple protocols.

These metrics fall into the following categories:

1. Base derived metrics
2. Metrics used for triage
3. Metrics used for network diagnostics
4. Metrics used for session classification
5. Metrics used for end user performance optimization

It must be understood that when a metric is discussed, it includes the average, median, and other statistical variations of that metric.

In the next section, we will discuss the base metrics. In later sections, we will discuss the more advanced metrics and their uses.

## 8 Base Derived Metrics

The base metrics which may be derived from the PDM are:

1. One-way delay
2. Round-trip delay
3. Server delay

### 8.1 One-Way Delay

One-way delay is the time taken to traverse the path one way between one network device to another. The path from A to B is distinguished from the path from B to A. For many reasons, the paths may have different characteristics and may have different delays. One-way delay is discussed in "A One-way Delay Metric for IPPM" [RFC2679].

### 8.2 Round-Trip Delay

Round-trip delay is the time taken to traverse the path both ways between one network device to another. The entire delay to travel

from A to B and B to A is used. Round-trip delay cannot tell if one path is quite different from another. Round-trip delay is discussed in "A Round-trip Delay Metric for IPPM" [RFC2681].

### 8.3 Server Delay

Server delay is the interval between when a packet is received by a device and a subsequent packet is sent back in response. This may be "Server Processing Time". It may also be a delay caused by acknowledgements. Server processing time includes the time taken by the combination of the stack and application to return the response.

## 9 Sample Implementation Flow (PDM Type 1)

Following is a sample simple flow with one packet sent from Host A and one packet received by Host B.

Time synchronization is required between Host A and Host B. See draft-ackermann-ntp-pdm-ntp-usage-00 [NTPPDM] for a description of how an NTP implementation may be set up to achieve good time synchronization.

Each packet, in addition to the PDM, contains information on the sender and receiver. This is the 5-tuple consisting of:

SADDR : IP address of the sender  
SPORT : Port for sender  
DADDR : IP address of the destination  
DPORT : Port for destination  
PROTC : Protocol for upper layer (ex. TCP, UDP, ICMP, etc.)

It should be understood that the packet identification information is in each packet. We will not repeat that in each of the following steps.

### 9.1 Step 1 (PDM Type 1)

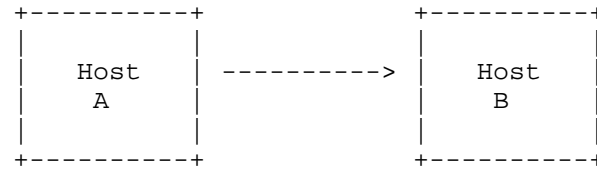
Packet 1 is sent from Host A to Host B. The time for Host A is set initially to 10:00AM.

The timestamp and packet sequence number are sent in the PDM.

The initial PSNTP from Host A starts at a random number. In this case, 25. The sub-second portion of the timestamp has been omitted for the sake of simplicity.



Packet 1



PDM Contents:

```

PSNTP : Packet Sequence Number This Packet:   25
TSTP  : Timestamp This Packet:                 10:00:00
PSNLR : Packet Sequence Number Last Received: -
TSLR  : Timestamp Last Received:                -
  
```

There are no derived statistics after packet 1.

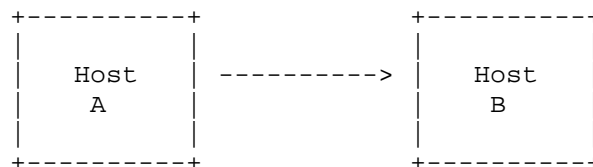
## 9.2 Step 2 (PDM Type 1)

Packet 1 is received by Host B. The time for Host B was synchronized with Host A. Both were set initially to 10:00AM.

The timestamp and PSN for the received packet are placed in the PSNLR and TSLR fields. These are from the point of view of B. That is, they indicate when the packet from A was received and which packet it was.

The PDM is not sent at this point. It is only prepared. It will be sent when the response to packet 1 is sent by Host B.

Packet 1 Received



PDM Contents:

```

PSNTP : Packet Sequence Number This Packet:   -
TSTP  : Timestamp This Packet:                 -
PSNLR : Packet Sequence Number Last Received: 25
TSLR  : Timestamp Last Received:                10:00:03
  
```

At this point, the following metric may be derived: one-way delay. In fact, we now know the one-way delay and the path. We will call this

path 1. This will be the outbound path from the point of view of Host A and the inbound path from the point of view of Host B.

The calculation of one-way delay (path 1) is as follows:

One-way delay (path 1) = Time packet 1 was received by B - Time Packet 1 was sent by A

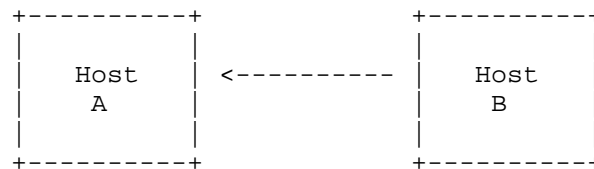
If we make the substitutions from our sample case above, then:

One-way delay (path 1) = 10:00:03 - 10:00:00 or 3 seconds

### 9.3 Step 3 (PDM Type 1)

Packet 2 is sent from Host B to Host A. The initial PSNTP from Host B starts at a random number. In this case, 12.

Packet 2



PDM Contents:

```

PSNTP : Packet Sequence Number This Packet: 12
TSTP  : Timestamp This Packet: 10:00:07
PSNLR : Packet Sequence Number Last Received: 25
TSLR  : Timestamp Last Received: 10:00:03
  
```

After Packet 2 is sent, the following metric may be derived: server delay.

The calculation of server delay is as follows:

Server delay = Time Packet 2 is sent by B - Time Packet 1 was received by B

Again, making the substitutions from the sample case: Server delay = 10:00:07 - 10:00:03 or 4 seconds

Further elaborations of server delay may be done by limiting the data length to be greater than 1. Some protocols, for example, TCP, have acknowledgements with a data length of 0 or keep-alive packets with a data length of 1. An ACK may precede the actual response data

packet. Keep-alives may be interspersed within the data flow.

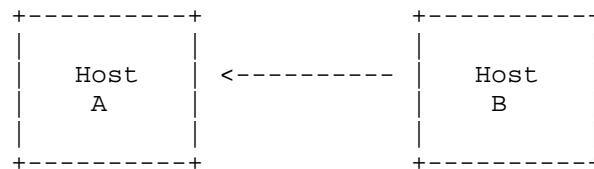
#### 9.4 Step 4 (PDM Type 1)

Packet 2 is received by Host A.

The timestamp and PSN for the received packet are placed in the PSNLR and TSLR fields. These are from the point of view of A. That is, they indicate when the packet from B was received and which packet it was.

The PDM is not sent at this point. It is only prepared. It will be sent when the NEXT packet to Host B is sent by Host A.

Packet 2 Received



PDM Contents:

```

PSNTP : Packet Sequence Number This Packet:  -
TSTP  : Timestamp This Packet:                -
PSNLR : Packet Sequence Number Last Received: 12
TSLR  : Timestamp Last Received:               10:00:10
  
```

However, at this point, the following metric may be derived: one-way delay (path 2).

The calculation of one-way delay (path 2) is as follows:

One-way delay (path 2) = Time packet 2 received by A - Time packet 2 sent by B

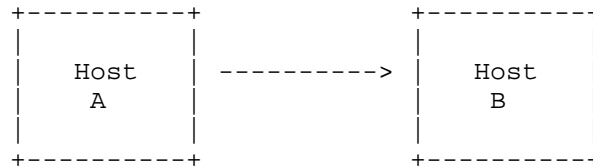
If we make the substitutions from our sample case above, then:

One-way delay (path 2) = 10:00:10 - 10:00:07 or 3 seconds

### 9.5 Step 5 (PDM Type 1)

Packet 3 is sent from Host A to Host B.

Packet 3



PDM Contents:

```
PSNTP : Packet Sequence Number This Packet: 26
TSTP  : Timestamp This Packet: 10:00:50
PSNLR : Packet Sequence Number Last Received: 12
TSLR  : Timestamp Last Received: 10:00:10
```

At this point the PDM flows across the network revealing the last received timestamp and PSN.

## 10 Sample Implementation Flow (PDM 2)

Following is a sample simple flow for PDM type 2 with one packet sent from Host A and one packet received by Host B. PDM type 2 does not require time synchronization between Host A and Host B. The calculations to derive meaningful metrics for network diagnostics is shown below each packet sent or received.

Each packet, in addition to the PDM contains information on the sender and receiver. As discussed before, a 5- tuple consists of:

```
SADDR : IP address of the sender
SPORT : Port for sender
DADDR : IP address of the destination
DPORT : Port for destination
PROTC : Protocol for upper layer (ex. TCP, UDP, ICMP)
```

It should be understood that the packet identification information is in each packet. We will not repeat that in each of the following steps.

### 10.1 Step 1 (PDM Type 2)

Packet 1 is sent from Host A to Host B. The time for Host A is set initially to 10:00AM.

The timestamp and packet sequence number are noted by the sender internally. The packet sequence number and timestamp are sent in the packet.

Packet 1



PDM type 2 Contents:

```

PSNTP   : Packet Sequence Number This Packet:    25
PSNLR   : Packet Sequence Number Last Received:  -
DELTALR : Delta Last Received:                   -
PSNLS   : Packet Sequence Number Last Sent:      -
DELTALS : Delta Last Sent:                       -
  
```

Internally, within the sender, Host A, it must keep:

```

PSNTP : Packet Sequence Number This Packet:    25
TSTP  : Timestamp This Packet:                 10:00:00
  
```

Note, the initial PSNTP from Host A starts at a random number. In this case, 25. The sub-second portion of the timestamp has been omitted for the sake of simplicity.

There are no derived statistics after packet 1.

## 10.2 Step 2 (PDM Type 2)

Packet 1 is received at Host B. His time is set to one hour later than Host A. In this case, 11:00AM

Internally, within the receiver, Host B, it must keep:

```

PSNLR : Packet Sequence Number Last Received:    25
TSLR  : Timestamp Last Received                  :    11:00:03
  
```

Note, this timestamp is in Host B time. It has nothing whatsoever to do with Host A time.

At this point, we have no derived statistics. In PDM type 1, the derived statistic one-way delay (path 1) could have been calculated. In PDM type 2, this is not possible because there is no time synchronization.

### 10.3 Step 3 (PDM Type 2)

Packet 2 is sent by Host B to Host A. Note, the initial PSNTP from Host B starts at a random number. In this case, 12. Before sending the packet, Host B does a calculation of deltas. Since Host B knows when it is sending the packet, and it knows when it received the previous packet, it can do the following calculation:

Sending time (packet 2) - receive time (packet 1)

We will call the result of this calculation: Delta Last Received.

That is:

DELTALR = Sending time (packet 2) - receive time (packet 1)

Note, both sending time and receive time are saved internally in Host B. They do not travel in the packet. Only the Delta is in the packet.

Assume that within Host B is the following:

```

PSNLR : Packet Sequence Number Last Received: 25
TSLR  : Timestamp Last Received           : 11:00:03
PSNTP : Packet Sequence Number This Packet : 12
TSTP  : Timestamp This Packet              : 11:00:07

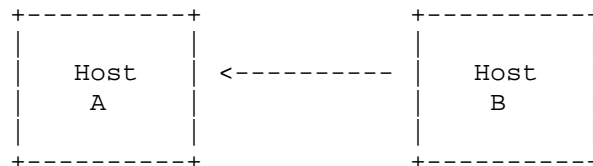
```

Hence, DELTALR becomes:

4 seconds = 11:00:07 - 11:00:03

Let us look at the PDM, and then we will look at the derived metrics at this point.

Packet 2



## PDM Type 2 Contents:

PSNTP	: Packet Sequence Number This Packet:	12
PSNLR	: Packet Sequence Number Last Received:	25
DELTALR	: Delta Last Received:	4
PSNLS	: Packet Sequence Number Last Sent:	-
DELTALS	: Delta Last Sent:	-

After Packet 2, the following metrics may be derived:

Server delay = DELTALR

Metrics left to be calculated are the path delay for path 2. This may be calculated when Packet 3 is sent. Clearly, if there is NO next packet for the 5-tuple, then this value will be missing.

## 10.4 Step 4 (PDM Type 2)

Packet 2 is received at Host A. Remember, its time is set to one hour earlier than Host B. It will keep internally:

PSNLR	: Packet Sequence Number Last Received:	12
TSLR	: Timestamp Last Received	: 10:00:12

Note, this timestamp is in Host A time. It has nothing whatsoever to do with Host B time.

At this point, we have two derived metrics:

1. Two-way delay or Round Trip time
2. Total end-to-end time

The formula for end-to-time is:

Time Last Received - Time Last Sent

For example, packet 25 was sent by Host A at 10:00:00. Packet 12 was received by Host A at 10:00:12 so:

End-to-End response time = 10:00:12 - 10:00:00 or 12

This derived metric we will call DELTALS or Delta Last Sent.

To calculate two-way delay, the formula is:

Two-way delay = DELTALS - DELTALR

Or:

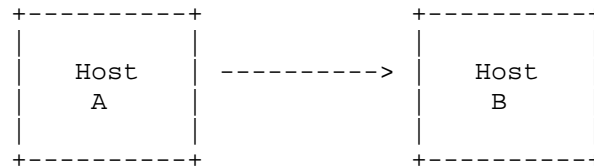
Two-way delay = 12 - 4 or 8

Now, the only problem is that at this point all metrics are in the Host and not exposed in a packet. To do that, we need a third packet.

#### 10.5 Step 5 (PDM Type 2)

Packet 3 is sent from Host A to Host B.

Packet 3



PDM Type 2 Contents:

PSNTP	: Packet Sequence Number This Packet:	26
PSNLR	: Packet Sequence Number Last Received:	12
DELTALR	: Delta Last Received:	*
PSNLS	: Packet Sequence Number Last Sent:	25
DELTALS	: Delta Last Sent:	12

#### 11 Derived Metrics : Advanced

A number of more advanced metrics may be derived from the data contained in the PDM. Some are relationships between two packets, others require analysis of multiple packets. The more advanced metrics fall into the categories shown below:

1. Metrics used for triage
2. Metrics used for network diagnostics
3. Metrics used for session classification
4. Metrics used for end user performance optimization

We will discuss each of these in turn.

##### 11.1 Advanced Derived Metrics : Triage

In this case, triage means to distinguish between problems occurring on the network paths or the server. The PDM provides one-way delay and server delay. This will enable distinguishing which path is a bottleneck as well as whether the server is a bottleneck.



## 11.2 Advanced Derived Metrics : Network Diagnostics

The data provided by the PDM may be used in combination with data fields in other protocols. We will call this Inter-Protocol Network Diagnostics (IPND).

The PDM also allows us to use only a single trace point for a number of diagnostic situations where today we need to trace at multiple points to get required data. In diagnostics, there is often the question of did the end device really send the packet and it got lost in the network or did it not send it at all.

So, what is done is that diagnostic traces are run at both client and server to get the required data. With the data provided by the PDM, in a number of the cases, this will not be necessary.

For example, taking PDM values along with data fields in the TCP protocol, the following may be found:

1. Retransmit duplication (RD)
2. ACK lag (AL)
3. Third-party connection reset (TPCR)
4. Elapsed time connection reset (ETCR)

A description of these follows.

### 11.2.1 Retransmit Duplication (RD)

The TCP protocol will retransmit segments given indications from the partner that it has not received them. The retransmitted segments contain the TCP sequence number and acknowledgement. The sequence number is started at a random number and increased by the amount of data sent in each packet.

Consider the following scenario. There is a packet sequence number in the packet at the IP layer. This is in the PDM that we have defined. The TCP sequence number already exists in the protocol.

Host A sends the following packets:

IP PSN 20, TCP SEQ 10  
IP PSN 21, TCP SEQ 11  
IP PSN 22, TCP SEQ 12

Host B receives:

IP PSN 20, TCP SEQ 10  
IP PSN 22, TCP SEQ 12

Host B indicates to Host A to resend packet with TCP SEQ 2.  
Retransmits are done at the TCP layer.

Host A sends the following packet:

IP PSN 23, TCP SEQ 11

The packet never reaches B. B waits until a timeout for retransmits expires. It asks for the packet again.

Host A sends the following packet:

IP PSN 24, TCP SEQ 11

This time, it reaches Host B. Having the combination of PSN (as provided in the PDM) and the TCP sequence number allows us to see whether the problem is that the network is losing the packet or somehow, the sender is not sending the packet correctly.

As we said before, this also allows us a single trace point rather than at the client and server to get the required data.

#### 11.2.2 ACK Lag (AL)

Some protocols, such as TCP, acknowledge packets. The PDM will allow or a calculation of rate of ACKs. Clients can be reconfigured to optimize acknowledgements and to speed traffic flow.

#### 11.2.3 Third-party Connection Reset (TPCR)

Connections may be aborted by a packet containing a particular flag. In the TCP protocol, this is the RESET flag. Sometimes a third-party, for example, a VPN router, will abort the connection. This may happen because the router is overloaded, the traffic is too noisy, or other reasons. This can also be quite hard to detect because the third-party will spoof the address of the sender.

Much time can be spent by the two endpoints pointing fingers at the other for having dropped the connection.

Such a third-party spoofer would likely not have the PDM Destination Option. Routers and other middle boxes are not required to support the Destination Options Extension Header. Even if a PDM DOH was generated, it would most likely violate the pattern of PSNs and time stamps being used. This would be a clue to the diagnostician that the TPCR event has occurred.

#### 11.2.4 Potential Hang (PH)

Connections may be aborted by a packet containing a particular flag. In the TCP protocol, this is the RESET flag. Sometimes this is done because a set amount of time has elapsed without activity. The PSN in the PDM can be used to determine the last packet sent by the partner and if a response is required -- a "hang" situation.

This can be distinguished from connections which are set to be aborted after a certain period of inactivity.

#### 11.3 Advanced Metrics : Session Classification

The PDM may be used to classify sessions as follows:

- One way traffic flow
- Two way traffic flow
- One way traffic flow with keep-alive
- Two way traffic flow with keep-alive
- Multiple send traffic flow
- Multiple receive traffic flow
- Full duplex traffic flow
- Half duplex traffic flow

- Immediate ACK data flow
- Delayed ACK data flow
- Proxied ACK data flow

A session classification system will assist the network diagnostician. This system will also help in categorizing the server delay.

#### 12 Use Cases

The scheme outlined above can also handle the following types of cases:

1. Host clocks not synchronized (shown above)
2. IP fragmentation
3. Multiple sends from one side (multiple segments)
4. Out of order segments
5. Retransmits
6. One-way transmit only (ex. FTP)
7. One-way transmit only  
(e.g. real time transports and streaming protocols)
8. Duplicate ACKs
9. Duplicate segments
10. Delayed ACKs

11. ACKs preceeding send for another reason
12. Proxy servers
13. Full duplex traffic
14. Keep alive (0 / 1 byte segments, larger segments)
15. No response from other side
16. Drop without retransmit (real time transports)
17. Looped packets (where the same packet may pass the same point multiple times without duplication)
18. Multihoming via SHIM6

### 13 Security Considerations

There are no security considerations.

### 14 IANA Considerations

There are no IANA considerations.

### 15 References

#### 15.1 Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1323] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander,

"SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,  
"Network Time Protocol Version 4: Protocol and Algorithms  
Specification", RFC 5905, June 2010.

## 15.2 Informative References

[NTPPDM] Ackermann, M., "draft-ackermann-ntp-pdm-ntp-usage-00",  
Internet Draft, January 2014.

[ELKPDM] Elkins, N., "draft-elkins-6man-ipv6-pdm-dest-option-05",  
Internet Draft, January 2014.

[IEEE1588] IEEE 1588-2002 standard, "Standard for a Precision Clock  
Synchronization Protocol for Networked Measurement and Control  
Systems"

## 16 Acknowledgments

The authors would like to thank Al Morton, Brian Trammel, David  
Boyes, and Rick Troth for their comments and assistance.

## Authors' Addresses

Nalini Elkins  
Inside Products, Inc.  
36A Upper Circle  
Carmel Valley, CA 93924  
United States  
Phone: +1 831 659 8360  
Email: [nalini.elkins@insidethestack.com](mailto:nalini.elkins@insidethestack.com)  
<http://www.insidethestack.com>

William Jouris  
Inside Products, Inc.  
36A Upper Circle  
Carmel Valley, CA 93924  
United States  
Phone: +1 925 855 9512  
Email: [bill.jouris@insidethestack.com](mailto:bill.jouris@insidethestack.com)  
<http://www.insidethestack.com>

Michael S. Ackermann  
Blue Cross Blue Shield of Michigan  
P.O. Box 2888  
Detroit, Michigan 48231  
United States

Phone: +1 310 460 4080  
Email: mackermann@bcbsmi.com  
<http://www.bcbsmi.com>

Keven Haining  
US Bank  
16900 W Capitol Drive  
Brookfield, WI 53005  
United States  
Phone: +1 262 790 3551  
Email: keven.haining@usbank.com  
<http://www.usbank.com>

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 17, 2014

P. Fan  
China Mobile  
February 13, 2014

Performance Metrics for Web Browsing  
draft-fan-ippm-web-metrics-00

Abstract

This document specifies metrics to evaluate performance for web browsing service.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 17, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Metrics . . . . .	2
2.1. Host Connection Delay . . . . .	2
2.2. Element Request Delay . . . . .	2
2.3. Element Waiting Delay . . . . .	3
2.4. Element Receiving Delay . . . . .	3
2.5. Element Loading Success . . . . .	3
2.6. Web Page TTFB (Time To First Byte) . . . . .	4
2.7. Web Page Loading Time . . . . .	4
3. Security Considerations . . . . .	4
4. IANA Considerations . . . . .	4
5. Normative References . . . . .	4
Author's Address . . . . .	4

## 1. Introduction

Web browsing has become a fundamental service in today's internet. With its massive users, web browsing service has contributed a large proportion of the total network traffic. Understanding how network performance influences web browsing experience benefits both network and content providers, and measurements like web page loading test are frequently conducted in practice. However, there is currently no standard metric to measure such performance. This document intends to add metrics for web browsing to the set of IP Performance Metrics (IPPM).

## 2. Metrics

This section gives description of a list of metrics that are used to evaluate web browsing related performance.

## 2.1. Host Connection Delay

Host Connect Delay is the time required to create a TCP connection to the web server. If a secure HTTPS connection is being used this time includes the SSL handshake process. The value of a Host Connection Delay is either a real number, or an undefined (informally, infinite) number of seconds. Note that Keep-Alive connections are often used to avoid the overhead of repeatedly connecting to the web server, so this delay is not always necessarily before loading every element.

## 2.2. Element Request Delay

Element Request Delay indicates the time required to send the HTTP request message to the server. The value of an Element Request Delay is either a real number, or an undefined (informally, infinite)



number of seconds. An Element Request Delay for an element from a web client is the time from the point when the client starts to send the HTTP request message for the element to the point when the client finishes sending the HTTP request message. This time will depend on the amount of data that is sent to the server. For example, long Send times will result from uploading files using an HTTP POST method.

### 2.3. Element Waiting Delay

Element Waiting Delay indicates the idle time spent waiting for a response message from the server. The value of an Element Waiting Delay is either a real number, or an undefined (informally, infinite) number of seconds. An Element Waiting Delay for an element from a web client is the time from the point when the client finishes sending the HTTP request message to the point when the client receives the first byte of the HTTP response message. This value will depend on the delays introduced due to network latency and the time required to process the request on the web server.

### 2.4. Element Receiving Delay

Element Receiving Delay indicates the time taken to read the HTTP response message from the web server. The value of an Element Receiving Delay is either a real number, or an undefined (informally, infinite) number of seconds. An Element Receiving Delay for an element from a web client is the time from the point when the client receives the first byte of the HTTP response message for the element to the point when the client finishes receiving the HTTP response message. This value will depend on the size of the content returned and network bandwidth.

### 2.5. Element Loading Success

Element Loading Success indicates the result of the HTTP transaction with the web server to download a web page element. The value of an Element Loading Success is either a one (signifying successful loading of the element) or a zero (signifying unsuccessful loading of the element). An Element Loading Success for an element from a web client is 1 exactly when the Element Request Delay, Element Waiting Delay and Element Receiving Delay are all a finite value; An Element Loading Success for an element from a web client is 0 exactly when any of the Element Request Delay, Element Waiting Delay and Element Receiving Delay is undefined.

## 2.6. Web Page TTFB (Time To First Byte)

Web Page TTFB indicates the duration needed to receive the first byte from the web server when loading a web page. The value of a Web Page TTFB is either a real number, or an undefined (informally, infinite) number of seconds. A Web Page TTFB for a web page from a web client is the time from the point when the client starts to send the first HTTP request message to the point when the client receives the first byte of the first HTTP response message.

## 2.7. Web Page Loading Time

Web Page Loading Time indicates the duration needed to receive all the elements from the web server when loading a web page. The value of a Web Page Loading Time is either a real number, or an undefined (informally, infinite) number of seconds. A Web Page Loading Time for a web page from a web client is the time from the point when the client starts to send the first HTTP request message to the point when the client finishes receiving the last HTTP response message.

## 3. Security Considerations

TBD.

## 4. IANA Considerations

The document makes no request for IANA action at this time.

## 5. Normative References

- [RFC2330] Paxson, V., Almes, G., and J. Mahdavi, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2616] Fielding, R., Gettys, J., and J. Mogul, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

## Author's Address

Peng Fan  
China Mobile  
32 Xuanwumen West Street, Xicheng District  
Beijing 100053  
P.R. China

Email: fanpeng@chinamobile.com

Network Working Group  
Internet-Draft  
Updates: 2330 (if approved)  
Intended status: Informational  
Expires: November 29, 2014

J. Fabini  
Vienna University of Technology  
A. Morton  
AT&T Labs  
May 28, 2014

Advanced Stream and Sampling Framework for IPPM  
draft-ietf-ippm-2330-update-05

Abstract

To obtain repeatable results in modern networks, test descriptions need an expanded stream parameter framework that also augments aspects specified as Type-P for test packets. This memo updates the IP Performance Metrics (IPPM) Framework RFC 2330 with advanced considerations for measurement methodology and testing. The existing framework mostly assumes deterministic connectivity, and that a single test stream will represent the characteristics of the path when it is aggregated with other flows. Networks have evolved and test stream descriptions must evolve with them, otherwise unexpected network features may dominate the measured performance. This memo describes new stream parameters for both network characterization and support of application design using IPPM metrics.

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 November 29, 2014.

## Copyright Notice

Copyright (c) 2014 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. Definition: Reactive Path Behavior . . . . .	4
2. Scope . . . . .	4
3. New or Revised Stream Parameters . . . . .	5
3.1. Test Packet Type-P . . . . .	6
3.1.1. Multiple Test Packet Lengths . . . . .	6
3.1.2. Test Packet Payload Content Optimization . . . . .	7
3.2. Packet History . . . . .	8
3.3. Access Technology Change . . . . .	8
3.4. Time-Slotted Randomness Cancellation . . . . .	8
4. Quality of Metrics and Methodologies . . . . .	10
4.1. Revised Definition of Repeatability . . . . .	10
4.2. Continuity No Longer an Alternative Repeatability Criterion . . . . .	11
4.3. Metrics Should Be Actionable . . . . .	12
4.4. It May Not Be Possible To Be Conservative . . . . .	12
4.5. Spatial and Temporal Composition Support Unbiased Sampling . . . . .	13
4.6. When to Truncate the Poisson Sampling Distribution . . . . .	13
5. Conclusions . . . . .	13
6. Security Considerations . . . . .	14
7. IANA Considerations . . . . .	14
8. Acknowledgements . . . . .	14
9. References . . . . .	14
9.1. Normative References . . . . .	14
9.2. Informative References . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

The IETF IP Performance Metrics (IPPM) working group first created a framework for metric development in [RFC2330]. This framework has stood the test of time and enabled development of many fundamental metrics, while only being updated once in a specific area [RFC5835].

The IPPM framework [RFC2330] generally relies on several assumptions, one of which is not explicitly stated but assumed: lightly loaded paths conform to the linear "delay = packet size / capacity" equation, being state/history-less (with some exceptions, firewalls are mentioned). However, this does not hold true for many modern network technologies, such as reactive paths (those with demand-driven resource allocation) and links with time-slotted operation. Per-flow state can be observed on test packet streams, and such treatment will influence network characterization if it is not taken into account. Flow history will also affect the performance of applications and be perceived by their users.

Moreover, Sections 4 and 6.2 of [RFC2330] explicitly recommend repeatable measurement metrics and methodologies. Measurements in today's access networks illustrate that methodological guidelines of [RFC2330] must be extended to capture the reactive nature of these networks. There are proposed extensions to allow methodologies to fulfill the continuity requirement stated in section 6.2 of [RFC2330], but is impossible to guarantee they can do so. Practical measurements confirm that some link types exhibit distinct responses to repeated measurements with identical stimulus, i.e., identical traffic patterns. If feasible, appropriate fine-tuning of measurement traffic patterns can improve measurement continuity and repeatability for these link types as shown in [IBD].

This memo updates the IP Performance Metrics (IPPM) Framework [RFC2330] with advanced considerations for measurement methodology and testing. We note that the scope of IPPM work at the time of [RFC2330] publication (and more than a decade of work that followed) was limited to active techniques, or those which generate packet streams which are dedicated to measurement and do not monitor user traffic. This memo retains that same scope.

We stress that this update of [RFC2330] does not invalidate or require changes to the analytic metric definitions prepared in the IPPM working group to date. Rather, it adds considerations for active measurement methodologies and expands the importance of existing conventions and notions in [RFC2330], such as "packets of Type-P".

Among the evolutionary networking changes is a phenomenon we call "reactive behavior", defined below.

#### 1.1. Definition: Reactive Path Behavior

Reactive path behavior will be observable by the test packet stream as a repeatable phenomenon where packet transfer performance characteristics \*change\* according to prior observations of the packet flow of interest (at the reactive host or link). Therefore, reactive path behavior is nominally deterministic with respect to the flow of interest. Other flows or traffic load conditions may result in additional performance-affecting reactions, but these are external to the characteristics of the flow of interest.

In practice, a sender may not have absolute control of the ingress packet stream characteristics at a reactive host or link, but this does not change the deterministic reactions present there. If we measure a path, the arrival characteristics at the reactive host/link are determined by the sending characteristics and the transfer characteristics of intervening hosts and links. Identical traffic patterns at the sending host might generate distinct patterns at the reactive host's/link's input due to impairments in the intermediate subpath. The reactive host/link is expected to provide deterministic response on identical input patterns.

Other than the size of the payload at the layer of interest and the header itself, packet content does not influence the measurement. Reactive behavior at the IP layer is not influenced by the TCP ports in use, for example. Therefore, the indication of reactive behavior must include the layer at which measurements are instituted.

Examples include links with Active/In-active state detectors, and hosts or links that revise their traffic serving and forwarding rates (up or down) based on packet arrival history.

Although difficult to handle from a measurement point of view, reactive paths entities are usually designed to improve overall network performance and user experience, for example by making capacity available to an active user. Reactive behavior may be an artifact of solutions to allocate scarce resources according to the demands of users, thus it is an important problem to solve for measurement and other disciplines, such as application design.

#### 2. Scope

The purpose of this memo is to foster repeatable measurement results in modern networks by highlighting the key aspects of test streams

and packets and make them part of the IPPM performance metric framework.

The scope is to update key sections of [RFC2330], adding considerations that will aid the development of new measurement methodologies intended for today's IP networks. Specifically, this memo describes useful stream parameters in addition to the information in Section 11.1 of [RFC2330] and described in [RFC3432] for periodic streams.

The memo also provides new considerations to update the criteria for metrics in section 4 of [RFC2330], the measurement methodology in section 6.2 of [RFC2330], and other topics related to the quality of metrics and methods (see section 4).

Other topics in [RFC2330] which might be updated or augmented are deferred to future work. This includes the topics of passive and various forms of hybrid active/passive measurements.

### 3. New or Revised Stream Parameters

There are several areas where measurement methodology definition and test result interpretation will benefit from an increased understanding of the stream characteristics and the (possibly unknown) network condition that influence the measured metrics.

1. Network treatment depends on the fullest extent on the "packet of Type-P" definition in [RFC2330], and has for some time.
  - \* State is often maintained on the per-flow basis at various points in the path, where "flows" are determined by IP and other layers. Significant treatment differences occur with the simplest of Type-P parameters: packet length. Use of multiple lengths is RECOMMENDED.
  - \* Payload content optimization (compression or format conversion) in intermediate segments breaks the convention of payload correspondence when correlating measurements are made at different points in a path.
2. Packet history (instantaneous or recent test rate or inactivity, also for non-test traffic) profoundly influences measured performance, in addition to all the Type-P parameters described in [RFC2330].
3. Access technology may change during testing. A range of transfer capacities and access methods may be encountered during a test session. When different interfaces are used, the host seeking

access will be aware of the technology change which differentiates this form of path change from other changes in network state. Section 14 of [RFC2330] treats the possibility that a host may have more than one attachment to the network, and also that assessment of the measurement path (route) is valid for some length of time (in Section 5 and Section 7 of [RFC2330]). Here we combine these two considerations under the assumption that changes may be more frequent and possibly have greater consequences on performance metrics.

4. Paths including links or nodes with time-slotted service opportunities represent several challenges to measurement (when service time period is appreciable):
  - \* Random/unbiased sampling is not possible beyond one such link in the path.
  - \* The above encourages a segmented approach to end to end measurement, as described in [RFC6049] for Network Characterization (as defined in [RFC6703]) to understand the full range of delay and delay variation on the path. Alternatively, if application performance estimation is the goal (also defined in [RFC6703]), then a stream with un-biased or known-bias properties [RFC3432] may be sufficient.
  - \* Multi-modal delay variation makes central statistics unimportant, others must be used instead.

Each of these topics is treated in detail below.

### 3.1. Test Packet Type-P

We recommend two Type-P parameters to be added to the factors which have impact on path performance measurements, namely packet length and payload type. Carefully choosing these parameters can improve measurement methodologies in their continuity and repeatability when deployed in reactive paths.

#### 3.1.1. Multiple Test Packet Lengths

Many instances of network characterization using IPPM metrics have relied on a single test packet length. When testing to assess application performance or an aggregate of traffic, benchmarking methods have used a range of fixed lengths and frequently augmented fixed size tests with a mixture of sizes, or IMIX as described in [RFC6985].



Test packet length influences delay measurements, in that the IPPM one-way delay metric [RFC2679] includes serialization time in its first-bit to last bit time stamping requirements. However, different sizes can have a larger influence on link delay and link delay variation than serialization would explain alone. This effect can be non-linear and change the instantaneous network performance when a different size is used, or the performance of packets following the size change.

Repeatability is a main measurement methodology goal as stated in section 6.2 of [RFC2330]. To eliminate packet length as a potential measurement uncertainty factor, successive measurements must use identical traffic patterns. In practice a combination of random payload and random start time can yield representative results as illustrated in [IRR].

### 3.1.2. Test Packet Payload Content Optimization

The aim for efficient network resource use has resulted in deployment of server-only or client-server lossless or lossy payload compression techniques on some links or paths. These optimizers attempt to compress high-volume traffic in order to reduce network load. Files are analyzed by application-layer parsers, and parts (like comments) might be dropped. Although typically acting on HTTP or JPEG files, compression might affect measurement packets, too. In particular, measurement packets are qualified for efficient compression when they use standard plain-text payload. We note that use of transport layer encryption will counteract the deployment of network-based analysis and may reduce the adoption of payload optimizations, however.

IPPM-conforming measurements should add packet payload content as a Type-P parameter which can help to improve measurement determinism. Some packet payloads are more susceptible to compression than others, but optimizers in the measurement path can be out ruled by using incompressible packet payload. This payload content could be supplied by a pseudo-random sequence generator or by using part of a compressed file (e.g., a part of a ZIP compressed archive).

Optimization can go beyond the scope of one single data- or measurement stream. Many more client- or network-centric optimization technologies have been proposed or standardized so far, including Robust Header Compression (ROHC) and Voice over IP aggregation as presented for instance in [EEAW]. Where optimization is feasible and valuable, many more of these technologies may follow. As a general observation, the more concurrent flows an intermediate host treats and the longer the paths shared by flows are, the higher becomes the incentive of hosts to aggregate flows belonging to distinct sources. Measurements should consider this potential

additional source of uncertainty with respect to repeatability. Aggregation of flows in networking devices can, for instance, result in reciprocal timing and performance influence of these flows which may exceed typical reciprocal queueing effects by orders of magnitude.

### 3.2. Packet History

Recent packet history and instantaneous data rate influence measurement results for reactive links supporting on-demand capacity allocation. Measurement uncertainty may be reduced by knowledge of measurement packet history and total host load. Additionally, small changes in history, e.g., because of lost packets along the path, can be the cause of large performance variations.

For instance, delay in reactive 3G networks like High Speed Packet Access (HSPA) depends to a large extent on the test traffic data rate. The reactive resource allocation strategy in these networks affects the uplink direction in particular. Small changes in data rate can be the reason of more than 200% increase in delay, depending on the specific packet size. A detailed theoretical and practical analysis of RRC link transitions, which can cause such behavior in Universal Mobile Terrestrial System (UMTS) networks, is presented, e.g., in [RRC].

### 3.3. Access Technology Change

[RFC2330] discussed the scenario of multi-homed hosts. If hosts become aware of access technology changes (e.g., because of IP address changes or lower layer information) and make this information available, measurement methodologies can use this information to improve measurement representativeness and relevance.

However, today's various access network technologies can present the same physical interface to the host. A host may or may not become aware when its access technology changes on such an interface. Measurements for paths which support on-demand capacity allocation are therefore challenging, in that it is difficult to differentiate between access technology changes (e.g., because of mobility) and reactive path behavior (e.g., because of data rate change).

### 3.4. Time-Slotted Randomness Cancellation

Time-Slotted operation of path entities - interfaces, routers or links - in a network path is a particular challenge for measurements, especially if the time slot period is substantial. The central observation as an extension to Poisson stream sampling in [RFC2330] is that the first such time-slotted component cancels unbiased

measurement stream sampling. In the worst case, time-slotted operation converts an unbiased, random measurement packet stream into a periodic packet stream. Being heavily biased, these packets may interact with periodic behavior of subsequent time-slotted network entities[TSRC].

Time-slotted randomness cancellation (TSRC) sources can be found in virtually any system, network component or path, their impact on measurements being a matter of the order of magnitude when compared to the metric under observation. Examples of TSRC sources include but are not limited to system clock resolution, operating system ticks, time-slotted component or network operation, etc. The amount of measurement bias is determined by the particular measurement stream, relative offset between allocated time-slots in subsequent path entities, delay variation in these paths, and other sources of variation. Measurement results might change over time, depending on how accurately the sending host, receiving host, and time-slotted components in the measurement path are synchronized to each other and to global time. If path segments maintain flow state, flow parameter change or flow re-allocations can cause substantial variation in measurement results.

Practical measurements confirm that such interference limits delay measurement variation to a sub-set of theoretical value range. Measurement samples for such cases can aggregate on artificial limits, generating multi-modal distributions as demonstrated in [IRR]. In this context, the desirable measurement sample statistics differentiate between multi-modal delay distributions caused by reactive path behavior and the ones due to time-slotted interference.

Measurement methodology selection for time-slotted paths depends to a large extent on the respective viewpoint. End-to-end metrics can provide accurate measurement results for short-term sessions and low likelihood of flow state modifications. Applications or services which aim at approximating path performance for a short time interval (in the order of minutes) and expect stable path conditions should therefore prefer end-to-end metrics. Here stable path conditions refer to any kind of global knowledge concerning measurement path flow state and flow parameters.

However, if long-term forecast of time-slotted path performance is the main measurement goal, a segmented approach relying on measurement of sub-path metrics is preferred. Re-generating unbiased measurement traffic at any hop can help to reveal the true range of path performance for all path segments.

#### 4. Quality of Metrics and Methodologies

[RFC6808] proposes repeatability and continuity as one of the metric and methodology properties to infer on measurement quality. Depending mainly on the set of controlled measurement parameters, measurements repeated for a specific network path using a specific methodology may or may not yield repeatable results. Challenging measurement scenarios for adequate parameter control include wireless, reactive, or time-slotted networks as discussed earlier in this document. This section presents an expanded definition of "repeatability" beyond the definition in [RFC2330] and an expanded examination of the [RFC2330] concept of "continuity" and its limited applicability.

##### 4.1. Revised Definition of Repeatability

[RFC2330] defines repeatability in a general way:

"A methodology for a metric should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, the same measurements should result in the same measurements."

The challenge is to develop this definition further, such that it becomes an objective measurable criterion (and does not depend on the concept of continuity discussed below). Fortunately, this topic has been treated in other IPPM work. In BCP 176 [RFC6576], the criteria of equivalent results was agreed as the surrogate for interoperability when assessing metric RFCs for standards track advancement. The criteria of equivalence were expressed as objective statistical requirements for comparison across same implementations and independent implementations in the test plans specific to each RFC evaluated ([RFC2679] in the test plan of [RFC6808]).

The tests of [RFC6808] rely on nearly identical conditions to be present for analysis, but accept that these conditions cannot be exactly identical in the production network paths used. The test plans allow some correction factors to be applied (some statistical tests are hyper-sensitive to differences in the mean of distributions), and recognize the original findings of [RFC2330] regarding excess sample sizes.

One way to view the reliance on identical conditions is to view it as a challenge: how few parameters and path conditions need to be controlled and still produce repeatable methods/measurements?

Although the [RFC6808] test plan documented numerical criteria for equivalence, we cannot specify the exact numerical criteria for

repeatability \*in general\*. The process in the BCP [RFC6576] and statistics in [RFC6808] have been used successfully, and the numerical criteria to declare a metric repeatable should be agreed by all interested parties prior to measurement.

We revise the definition slightly, as follows:

A methodology for a metric should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, the methods should produce equivalent measurement results.

#### 4.2. Continuity No Longer an Alternative Repeatability Criterion

In the original framework [RFC2330], the concept of continuity was introduced to provide a relaxed criteria for judging repeatability, and was described in section 6.2 of [RFC2330] as follows:

"...a methodology for a given metric exhibits continuity if, for small variations in conditions, it results in small variations in the resulting measurements."

Although there are conditions where metrics may exhibit continuity, there are others where this criteria would fail for both user traffic and active measurement traffic. Consider link fragmentation, and the non-linear increase in delay when we increase packet size just beyond the limit of a single fragment. An active measurement packet would see the same delay increase when exceeding the fragment size.

The Bulk Transfer Capacity (BTC) [RFC3148] gives another example at bottom of page 2:

"There is also evidence that most TCP implementations exhibit non-linear performance over some portion of their operating region. It is possible to construct simple simulation examples where incremental improvements to a path (such as raising the link data rate) results in lower overall TCP throughput (or BTC) [Mat98]."

Clearly, the time-slotted network elements described in section 3.4 above also qualifies as a new exception to the ideal of continuity.

Therefore, we deprecate continuity as an alternate criterion on metrics, and prefer the more exact evaluation of repeatability instead.

#### 4.3. Metrics Should Be Actionable

The IP Performance Metrics Framework [RFC2330] includes usefulness as a metric criterion:

"...The metrics must be useful to users and providers in understanding the performance they experience or provide...".

When considering measurements as part of a maintenance process, evaluation of measurement results for a path under observation can draw attention to potential performance problems "somewhere" on the path. Anomaly detection is therefore an important phase and first step which already satisfies the usefulness criterion for many metrics.

This concept of usefulness can be extended, becoming a sub-set of what we refer to as "actionable" criterion in the following. We note that this is not the term from law.

Central to maintenance is the isolation of the root cause of reported anomalies down to a specific sub-path, link or host, and metrics should support this second step as well. While detection of path anomaly may be the result of an on-going monitoring process, the second step of cause isolation consists of specific, directed on-demand measurements on components and sub-paths. Metrics must support users in this directed search, becoming actionable:

Metrics must enable users and operators to understand path performance and SHOULD help to direct corrective actions when warranted, based on the measurement results.

Besides characterizing metrics, usefulness and actionable properties are also applicable to methodologies and measurements.

#### 4.4. It May Not Be Possible To Be Conservative

[RFC2330] adopts the term "conservative" for measurement methodologies for which:

"... the act of measurement does not modify, or only slightly modifies, the value of the performance metric the methodology attempts to measure."

It should be noted that this definition of "conservative" in the sense of [RFC2330] depends to a large extent on the measurement path's technology and characteristics. In particular, when deployed on reactive paths, sub-paths, links or hosts conforming to the definition in Section 1.1 of this document, measurement packets can

originate capacity (re)allocations. In addition, small measurement flow variations can result in other users on the same path perceiving significant variations in measurement results. Therefore:

It is not always possible for the method to be conservative.

#### 4.5. Spatial and Temporal Composition Support Unbiased Sampling

Concepts related to temporal and spatial composition of metrics in Section 9 of [RFC2330] have been extended in [RFC5835]. [RFC5835] defines multiple new types of metrics, including Spatial Composition, Temporal Aggregation, and Spatial Aggregation. So far, only the metrics for Spatial Composition have been standardized [RFC6049], providing the ability to estimate the performance of a complete path from subpath metrics. Spatial Composition aligns with the finding of [TSRC], that unbiased sampling is not possible beyond the first time-slotted link within a measurement path.

In cases where unbiased measurement for all segments of a path is not feasible due to the presence of a time-slotted link, restoring randomness of measurement samples when necessary is recommended as presented in [TSRC], in combination with Spatial Composition [RFC6049].

#### 4.6. When to Truncate the Poisson Sampling Distribution

Section 11.1.1 of [RFC2330] describes Poisson sampling, where the inter-packet send times have a Poisson distribution. A path element with reactive behavior sensitive to flow inactivity could change state if the random inter-packet time is too long.

It is recommended to truncate the tail of Poisson distribution when needed to avoid reactive element state changes.

Tail truncation has been used without issue to ensure that minimum sample sizes can be attained in a fixed test interval.

#### 5. Conclusions

Safeguarding repeatability as a key property of measurement methodologies is highly challenging and sometimes impossible in reactive paths. Measurements in paths with demand-driven allocation strategies must use a prototypical application packet stream to infer a specific application's performance. Measurement repetition with unbiased network and flow states (e.g., by rebooting measurement hosts) can help to avoid interference with periodic network behavior, randomness being a mandatory feature for avoiding correlation with network timing.

Inferring the path performance between one measurement session or packet stream and other sessions/streams with alternate characteristics is generally discouraged with reactive paths because of the huge set of global parameters which have influence on instantaneous path performance.

## 6. Security Considerations

The security considerations that apply to any active measurement of live paths are relevant here as well. See [RFC4656] and [RFC5357].

When considering privacy of those involved in measurement or those whose traffic is measured, the sensitive information available to potential observers is greatly reduced when using active techniques which are within this scope of work. Passive observations of user traffic for measurement purposes raise many privacy issues. We refer the reader to the privacy considerations described in the Large Scale Measurement of Broadband Performance (LMAP) Framework [I-D.ietf-lmap-framework], which covers active and passive techniques.

## 7. IANA Considerations

This memo makes no requests of IANA.

## 8. Acknowledgements

The authors thank Rudiger Geib, Matt Mathis, Konstantinos Pentikousis, and Robert Sparks for their helpful comments on this memo, Alissa Cooper and Kathleen Moriarty for suggesting ways to "update the update" for heightened privacy awareness and its consequences, and Ann Cerveney for her editorial review and comments that helped to improve readability overall.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.



- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

## 9.2. Informative References

- [EEAW] Pentikousis, K., Piri, E., Pinola, J., Fitzek, F., Nissilae, T., and I. Harjula, "Empirical Evaluation of VoIP Aggregation over a Fixed WiMAX Testbed", Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks and communities (TridentCom '08) <http://dl.acm.org/citation.cfm?id=1390599>, March 2008.
- [I-D.ietf-lmap-framework] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-05 (work in progress), May 2014.
- [IBD] Fabini, J., Karner, W., Wallentin, L., and T. Baumgartner, "The Illusion of Being Deterministic - Application-Level Considerations on Delay in 3G HSPA Networks", Lecture Notes in Computer Science, Springer, Volume 5550, 2009, pp 301-312, May 2009.

- [IRR] Fabini, J., Wallentin, L., and P. Reichl, "The Importance of Being Really Random: Methodological Aspects of IP-Layer 2G and 3G Network Delay Assessment", ICC'09 Proceedings of the 2009 IEEE International Conference on Communications, doi: 10.1109/ICC.2009.5199514, June 2009.
- [Mat98] Mathis, M., "Empirical Bulk Transfer Capacity", IP Performance Metrics Working Group report in Proceeding of the Forty Third Internet Engineering Task Force, Orlando, FL. <http://www.ietf.org/proceedings/98dec/slides/ippm-mathis-98dec.pdf>, December 1998.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.
- [RFC6808] Ciavattone, L., Geib, R., Morton, A., and M. Wieser, "Test Plan and Results Supporting Advancement of RFC 2679 on the Standards Track", RFC 6808, December 2012.
- [RFC6985] Morton, A., "IMIX Genome: Specification of Variable Packet Sizes for Additional Testing", RFC 6985, July 2013.
- [RRC] Peraelae, P., Barbuzzi, A., Boggia, G., and K. Pentikousis, "Theory and Practice of RRC State Transitions in UMTS Networks", IEEE Globecom 2009 Workshops doi: 10.1109/GLOCOMW.2009.5360763, November 2009.
- [TSRC] Fabini, J. and M. Abmayer, "Delay Measurement Methodology Revisited: Time-slotted Randomness Cancellation", IEEE Transactions on Instrumentation and Measurement doi:10.1109/TIM.2013.2263914, October 2013.

#### Authors' Addresses

Joachim Fabini  
Vienna University of Technology  
Gusshausstrasse 25/E389  
Vienna 1040  
Austria

Phone: +43 1 58801 38813  
Fax: +43 1 58801 38898  
Email: [Joachim.Fabini@tuwien.ac.at](mailto:Joachim.Fabini@tuwien.ac.at)  
URI: <http://www.tc.tuwien.ac.at/about-us/staff/joachim-fabini/>

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

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

IPPM WG  
Internet-Draft  
Updates: 4656, 5357 (if approved)  
Intended status: Standards Track  
Expires: February 27, 2016

K. Pentikousis, Ed.  
EICT  
E. Zhang  
Y. Cui  
Huawei Technologies  
August 26, 2015

IKEv2-derived Shared Secret Key for O/TWAMP  
draft-ietf-ippm-ipsec-11

Abstract

The One-way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP) security mechanisms require that both the client and server endpoints possess a shared secret. This document describes the use of keys derived from an IKEv2 security association (SA) as the shared key in O/TWAMP. If the shared key can be derived from the IKEv2 SA, O/TWAMP can support certificate-based key exchange, which would allow for more operational flexibility and efficiency. The key derivation presented in this document can also facilitate automatic key management.

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 February 27, 2016.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
3. Scope . . . . .	4
4. O/TWAMP Security . . . . .	4
4.1. O/TWAMP-Control Security . . . . .	4
4.2. O/TWAMP-Test Security . . . . .	6
4.3. O/TWAMP Security Root . . . . .	6
5. O/TWAMP for IPsec Networks . . . . .	7
5.1. Shared Key Derivation . . . . .	7
5.2. Server Greeting Message Update . . . . .	8
5.3. Set-Up-Response Update . . . . .	9
5.4. O/TWAMP over an IPsec tunnel . . . . .	10
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Acknowledgements . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

The One-way Active Measurement Protocol (OWAMP) [RFC4656] and the Two-Way Active Measurement Protocol (TWAMP) [RFC5357] can be used to measure network performance parameters such as latency, bandwidth, and packet loss by sending probe packets and monitoring their experience in the network. In order to guarantee the accuracy of network measurement results, security aspects must be considered. Otherwise, attacks may occur and the authenticity of the measurement results may be violated. For example, if no protection is provided, an adversary in the middle may modify packet timestamps, thus altering the measurement results.

According to [RFC4656] [RFC5357], the O/TWAMP security mechanism requires that endpoints (i.e. both the client and the server) possess a shared secret. In today's network deployments, however, the use of pre-shared keys is far from optimal. For example, in wireless infrastructure networks, certain network elements, which can be seen as the two endpoints from an O/TWAMP perspective, support

certificate-based security. For instance, consider the case in which one wants to measure IP performance between a E-UTRAN Evolved Node B (eNB) and Security Gateway (SeGW), both of which are 3GPP Long Term Evolution (LTE) nodes and support certificate mode and the Internet Key Exchange Protocol Version 2 (IKEv2).

The O/TWAMP security mechanism specified in [RFC4656] [RFC5357] supports the pre-shared key mode only, hindering large-scale deployment of O/TWAMP: provisioning and management of "shared secrets" for massive deployments consumes a tremendous amount of effort and is prone to human error. At the same time, recent trends point to wider IKEv2 deployment which, in turn, calls for mechanisms and methods that enable tunnel end-users, as well as operators, to measure one-way and two-way network performance in a standardized manner.

With IKEv2 widely deployed, employing shared keys derived from an IKEv2 security association (SA) can be considered a viable alternative through the method described in this document. If the shared key can be derived from the IKEv2 SA, O/TWAMP can support certificate-based key exchange and practically increase operational flexibility and efficiency. The use of IKEv2 also makes it easier to extend automatic key management.

In general, O/TWAMP measurement packets can be transmitted inside the IPsec tunnel, as it occurs with typical user traffic, or transmitted outside the IPsec tunnel. This may depend on the operator's policy and the performance evaluation goal, and is orthogonal to the mechanism described in this document. When IPsec is deployed, protecting O/TWAMP traffic in unauthenticated mode using IPsec is one option. Another option is to protect O/TWAMP traffic using the O/TWAMP layer security established using the Pre-Shared Key (PSK) derived from IKEv2 and bypassing the IPsec tunnel.

Protecting unauthenticated O/TWAMP control and/or test traffic via Authentication Header (AH) [RFC4302] or Encapsulating Security Payload (ESP) [RFC4303] cannot provide various security options, e.g. it cannot authenticate part of a O/TWAMP packet as mentioned in [RFC4656]. For measuring latency, a timestamp is carried in O/TWAMP test traffic. The sender has to fetch the timestamp, encrypt it, and send it. When the mechanism described in this document is used, partial authentication of O/TWAMP packets is possible and therefore the middle step can be skipped, potentially improving accuracy as the sequence number can be encrypted and authenticated before the timestamp is fetched. The receiver obtains the timestamp without the need for the corresponding decryption step. In such cases, protecting O/TWAMP traffic using O/TWAMP layer security but bypassing the IPsec tunnel has its advantages.

This document specifies a method for enabling network measurements between a TWAMP client and a TWAMP server. In short, the shared key used for securing TWAMP traffic is derived from IKEv2 [RFC7296]. TWAMP implementations signal the use of this method by setting IKEv2Derived (see Section 7). IKEv2-derived keys SHOULD be used instead of shared secrets when O/TWAMP is employed in a deployment using IKEv2. From an operations and management perspective [RFC5706], the mechanism described in this document requires that both the TWAMP Control-Client and Server support IPsec.

The remainder of this document is organized as follows. Section 4 summarizes O/TWAMP protocol operation with respect to security. Section 5 presents the method for binding TWAMP and IKEv2 for network measurements between the client and the server which both support IKEv2. Finally, Section 6 discusses the security considerations arising from the proposed mechanisms.

## 2. Terminology

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

## 3. Scope

This document specifies a method using keys derived from an IKEv2 security association (SA) as the shared key in O/TWAMP. O/TWAMP implementations signal the use of this method by setting IKEv2Derived (see Section 7).

## 4. O/TWAMP Security

Security for O/TWAMP-Control and O/TWAMP-Test are briefly reviewed in the following subsections.

### 4.1. O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on

- o Advanced Encryption Standard (AES) in Cipher Block Chaining (AES-CBC) for confidentiality
- o Hash-based Message Authentication Code (HMAC)-Secure Hash Algorithm1 (SHA1) truncated to 128 bits for message authentication

Three modes of operation are supported in the OWAMP-Control protocol: unauthenticated, authenticated, and encrypted. In addition to these modes, the TWAMP-Control protocol also supports a mixed mode, i.e.

the TWAMP-Control protocol operates in encrypted mode while TWAMP-Test protocol operates in unauthenticated mode. The authenticated, encrypted and mixed modes require that endpoints possess a shared secret, typically a passphrase. The secret key is derived from the passphrase using a password-based key derivation function PBKDF2 (PKCS#5) [RFC2898].

In the unauthenticated mode, the security parameters are left unused. In the authenticated, encrypted and mixed modes, the security parameters are negotiated during the control connection establishment.

Figure 1 illustrates the initiation stage of the O/TWAMP-Control protocol between a Control-Client and a Server. In short, the Control-Client opens a TCP connection to the Server in order to be able to send O/TWAMP-Control commands. The Server responds with a Server Greeting, which contains the Modes, Challenge, Salt, Count, and MBZ fields (see Section 3.1 of [RFC4656]). If the Control-Client preferred mode is available, the client responds with a Set-Up-Response message, wherein the selected Mode, as well as the KeyID, Token and Client initialization vector (IV) are included. The Token is the concatenation of a 16-octet Challenge, a 16-octet AES Session-key used for encryption, and a 32-octet HMAC-SHA1 Session-key used for authentication. The Token is encrypted using AES-CBC.

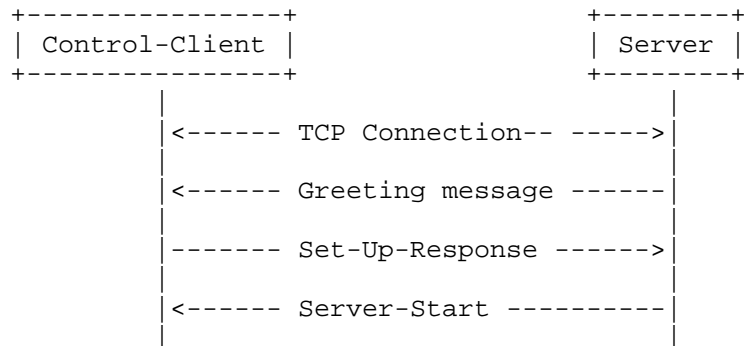


Figure 1: Initiation of O/TWAMP-Control

Encryption uses a key derived from the shared secret associated with KeyID. In the authenticated, encrypted and mixed modes, all further communication is encrypted using the AES Session-key and authenticated with the HMAC Session-key. After receiving the Set-Up-Response the Server responds with a Server-Start message containing the Server-IV. The Control-Client encrypts everything it transmits through the just-established O/TWAMP-Control connection using stream encryption with Client-IV as the IV. Correspondingly, the Server



encrypts its side of the connection using Server-IV as the IV. The IVs themselves are transmitted in cleartext. Encryption starts with the block immediately following that containing the IV.

The AES Session-key and HMAC Session-key are generated randomly by the Control-Client. The HMAC Session-key is communicated along with the AES Session-key during O/TWAMP-Control connection setup. The HMAC Session-key is derived independently of the AES Session-key.

#### 4.2. O/TWAMP-Test Security

The O/TWAMP-Test protocol runs over UDP, using the Session-Sender and Session-Reflector IP and port numbers that were negotiated during the Request-Session exchange. O/TWAMP-Test has the same mode with O/TWAMP-Control and all O/TWAMP-Test sessions inherit the corresponding O/TWAMP-Control session mode except when operating in mixed mode.

The O/TWAMP-Test packet format is the same in authenticated and encrypted modes. The encryption and authentication operations are, however, different. Similarly with the respective O/TWAMP-Control session, each O/TWAMP-Test session has two keys: an AES Session-key and an HMAC Session-key. However, there is a difference in how the keys are obtained:

O/TWAMP-Control: the keys are generated by the Control-Client and communicated to the Server during the control connection establishment with the Set-Up-Response message (as part of the Token).

O/TWAMP-Test: the keys are derived from the O/TWAMP-Control keys and the session identifier (SID), which serve as inputs to the key derivation function (KDF). The O/TWAMP-Test AES Session-key is generated using the O/TWAMP-Control AES Session-key, with the 16-octet session identifier (SID), for encrypting and decrypting the packets of the particular O/TWAMP-Test session. The O/TWAMP-Test HMAC Session-key is generated using the O/TWAMP-Control HMAC Session-key, with the 16-octet session identifier (SID), for authenticating the packets of the particular O/TWAMP-Test session.

#### 4.3. O/TWAMP Security Root

As discussed above, the O/TWAMP-Test AES Session-key and HMAC Session-key are derived, respectively, from the O/TWAMP-Control AES Session-key and HMAC Session-key. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the Control-Client, and encrypted with the shared secret associated with KeyID. Therefore, the security root is the shared

secret key. Thus, for large deployments, key provision and management may become overly complicated. Comparatively, a certificate-based approach using IKEv2 can automatically manage the security root and solve this problem, as we explain in Section 5.

## 5. O/TWAMP for IPsec Networks

This section presents a method of binding O/TWAMP and IKEv2 for network measurements between a client and a server which both support IPsec. In short, the shared key used for securing O/TWAMP traffic is derived using IKEv2 [RFC7296].

### 5.1. Shared Key Derivation

In the authenticated, encrypted and mixed modes, the shared secret key MUST be derived from the IKEv2 Security Association (SA). Note that we explicitly opt to derive the shared secret key from the IKEv2 SA, rather than the child SA, since it is possible that an IKEv2 SA is created without generating any child SA [RFC6023].

When the shared secret key is derived from the IKEv2 SA, SK\_d must be generated first. SK\_d must be computed as per [RFC7296].

The shared secret key MUST be generated as follows:

```
Shared secret key = prf( SK_d, "IPPM" )
```

Wherein the string "IPPM" is encoded in ASCII and "prf" is a pseudorandom function.

It is recommended that the shared secret key is derived in the IPsec layer so that IPsec keying material is not exposed to the O/TWAMP client. Note, however, that the interaction between the O/TWAMP and IPsec layers is host-internal and implementation-specific. Therefore, this is clearly outside the scope of this document, which focuses on the interaction between the O/TWAMP client and server. That said, one possible way could be the following: at the Control-Client side, the IPsec layer can perform a lookup in the Security Association Database (SAD) using the IP address of the Server and thus match the corresponding IKEv2 SA. At the Server side, the IPsec layer can look up the corresponding IKEv2 SA by using the Security Parameter Indexes (SPIs) sent by the Control-Client (see Section 5.3), and therefore extract the shared secret key.

In case that both client and server do support IKEv2 but there is no current IKEv2 SA, two alternative ways could be considered. First, the O/TWAMP Control-Client initiates the establishment of the IKEv2 SA, logs this operation, and selects the mode which supports IKEv2.

Alternatively, the O/TWAMP Control-Client does not initiate the establishment of the IKEv2 SA, logs an error for operational management purposes, and proceeds with the modes defined in [RFC4656][RFC5357][RFC5618]. Again, although both alternatives are feasible, they are in fact implementation-specific.

If rekeying for the IKEv2 SA or deletion of the IKEv2 SA occurs, the corresponding shared secret key generated from the SA MUST continue to be used until the O/TWAMP session terminates.

## 5.2. Server Greeting Message Update

To trigger a binding association between the key generated from IKEv2 and the O/TWAMP shared secret key, the Modes field in the Server Greeting Message (Figure 2) must support key derivation as discussed in Section 5.1. Support for deriving the shared key from the IKEv2 SA is indicated by setting IKEv2Derived (see Section 7). Therefore, when this method is used, the Modes value extension MUST be supported.

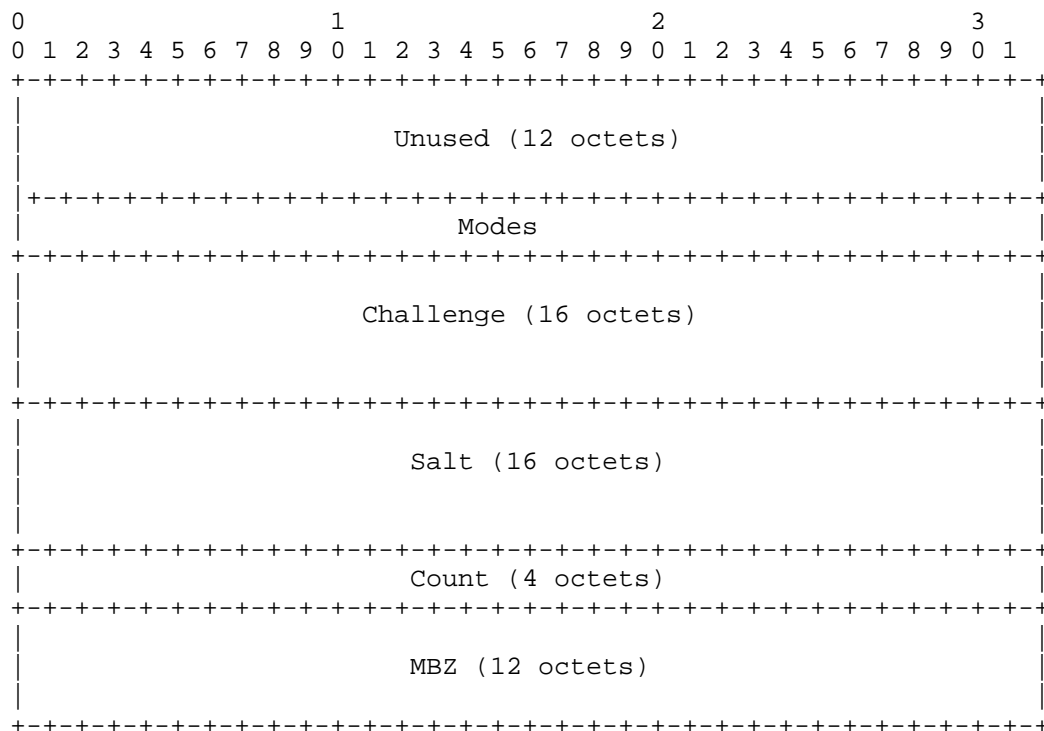


Figure 2: Server Greeting format

The choice of this set of Modes values poses no backwards compatibility problems to existing O/TWAMP clients. Robust legacy Control-Client implementations would disregard the fact that the IKEv2Derived Modes bit in the Server Greeting is set. On the other hand, a Control-Client implementing this method can identify that the O/TWAMP Server contacted does not support this specification. If the Server supports other Modes, as one could assume, the Control-Client would then decide which Mode to use and indicate such accordingly as per [RFC4656][RFC5357]. A Control-Client implementing this method which decides not to employ IKEv2 derivation, can simply behave as a purely [RFC4656]/[RFC5357] compatible client.

### 5.3. Set-Up-Response Update

The Set-Up-Response Message Figure 3 is updated as follows. When a O/TWAMP Control-Client implementing this method receives a Server Greeting indicating support for Mode IKEv2Derived it SHOULD reply to the O/TWAMP Server with a Set-Up response that indicates so. For example, a compatible O/TWAMP Control-Client choosing the authenticated mode with IKEv2 shared secret key derivation should set Mode bits as per Section 7.

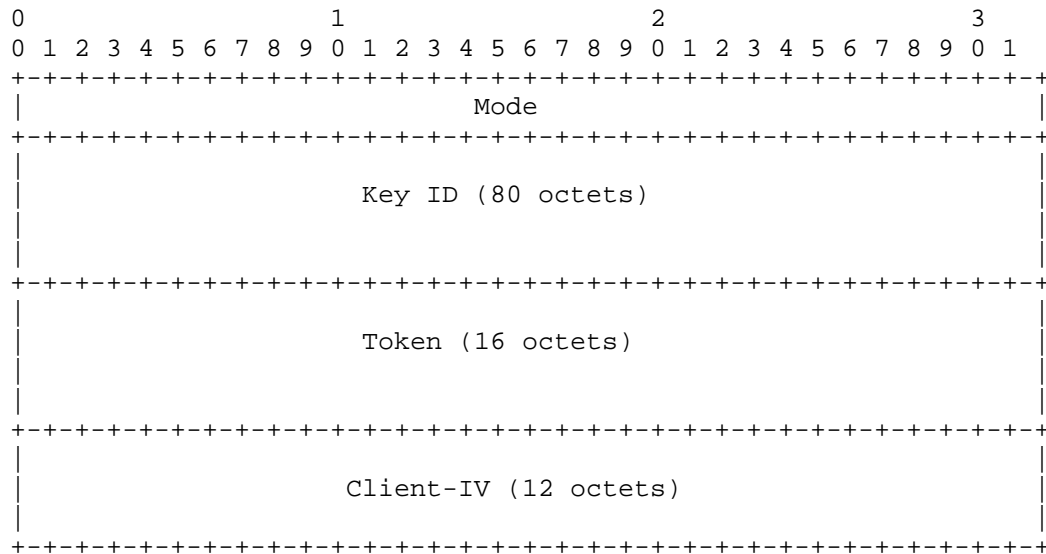


Figure 3: Set-Up-Response Message

The Security Parameter Index (SPI)(see [RFC4301] [RFC7296]) uniquely identifies the Security Association (SA). If the Control-Client supports IKEv2 SA shared secret key derivation, it will choose the corresponding Mode value and carry SPIi and SPIr in the Key ID field.

SPIi and SPIr MUST be included in the Key ID field of the Set-Up-Response Message to indicate the IKEv2 SA from which the O/TWAMP shared secret key derived from. The length of SPI is 8 octets. Therefore, the first 8 octets of Key ID field MUST carry SPIi and the second 8 octets MUST carry SPIr. The remaining bits of the Key ID field MUST be set to zero.

A O/TWAMP Server implementation MUST obtain the SPIi and SPIr from the first 16 octets and ignore the remaining octets of the Key ID field. Then, the Control-Client and the Server can derive the shared secret key based on the Mode value and SPI. If the O/TWAMP Server cannot find the IKEv2 SA corresponding to the SPIi and SPIr received, it MUST log the event for operational management purposes. In addition, the O/TWAMP Server SHOULD set the Accept field of the Server-Start message to the value 6 to indicate that the Server is not willing to conduct further transactions in this O/TWAMP-Control session since it can not find the corresponding IKEv2 SA.

#### 5.4. O/TWAMP over an IPsec tunnel

IPsec Authentication Header (AH) [RFC4302] and Encapsulating Security Payload (ESP) [RFC4303] provide confidentiality and data integrity to IP datagrams. An IPsec tunnel can be used to provide the protection needed for O/TWAMP Control and Test packets, even if the peers choose the unauthenticated mode of operation. In order to ensure authenticity and security, O/TWAMP packets between two IKEv2 systems SHOULD be configured to use the corresponding IPsec tunnel running over an external network even when using the O/TWAMP unauthenticated mode.

#### 6. Security Considerations

As the shared secret key is derived from the IKEv2 SA, the key derivation algorithm strength and limitations are as per [RFC7296]. The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator employed. The strength of all keys and implementation vulnerabilities, particularly Denial of Service (DoS) attacks are as defined in [RFC7296].

#### 7. IANA Considerations

During the production of this document, the authors and reviewers noticed that the TWAMP-Modes registry should describe a field of single bit position flags, rather than the current registry construction with assignment of integer values. In addition, the Semantics Definition column seems to have spurious information in it.

The registry should be re-formatted to simplify future assignments. Thus, the current contents of the TWAMP-Modes Registry should appear as follows:

Bit Pos	Description	Semantics Definition	Reference
0	Unauthenticated	Section 3.1	[RFC4656]
1	Authenticated	Section 3.1	[RFC4656]
2	Encrypted	Section 3.1	[RFC4656]
3	Unauth.TEST protocol,Encrypted CONTROL	Section 3.1	[RFC5618]
4	Individual Session Control		[RFC5938]
5	Reflect Octets Capability		[RFC6038]
6	Symmetrical Size Sender Test Packet Format		[RFC6038]

Figure 4: TWAMP Modes registry

The new description and registry management instructions follow.

**Registry Specification:** TWAMP-Modes are specified in TWAMP Server Greeting messages and Set-up Response messages consistent with section 3.1 of [RFC5357]. Modes are indicated by setting single bits in the 32-bit Modes Field.

**Registry Management:** Because the "TWAMP-Modes" are based on only 32 bit positions with each position conveying a unique feature, and because TWAMP is an IETF protocol, this registry must be updated only by "IETF Consensus" as specified in [RFC5226]. IANA SHOULD allocate monotonically increasing bit positions when requested.

**Experimental Numbers:** No experimental bit positions are currently assigned in the Modes Registry, as indicated in the initial contents above.

In addition, this document requests allocation of a new entry in the TWAMP-Modes registry:

Bit Pos	Description	Semantics Definition	Reference
---	-----	-----	-----
X	IKEv2Derived Mode Capability	Section 5	[RFCxxxx]

where IANA is requested to assign new bit position, X, and RFCxxxx refers to this memo when published.

Figure 5: TWAMP IKEv2-derived Mode Capability

For the new OWAMP-Modes Registry, see the IANA Considerations in [I-D.ietf-ippm-owamp-registry].

## 8. Acknowledgements

We thank Eric Chen, Yaakov Stein, Brian Trammell, Emily Bi, John Mattsson, Steve Baillargeon, Spencer Dawkins, Tero Kivinen, Fred Baker, Meral Shirazipour, Hannes Tschofenig, Ben Campbell, Stephen Farrell, Brian Haberman, and Barry Leiba for their reviews, comments and text suggestions.

Al Morton deserves a special mention for his thorough reviews and text contributions to this document as well as the constructive discussions over several IPPM meetings.

## 9. References

### 9.1. Normative References

- [I-D.ietf-ippm-owamp-registry] Morton, A., "Registries for the One-Way Active Measurement Protocol - OWAMP", draft-ietf-ippm-owamp-registry-00 (work in progress), July 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<http://www.rfc-editor.org/info/rfc4656>>.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<http://www.rfc-editor.org/info/rfc5357>>.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, DOI 10.17487/RFC5618, August 2009, <<http://www.rfc-editor.org/info/rfc5618>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.

## 9.2. Informative References

- [RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, DOI 10.17487/RFC2898, September 2000, <<http://www.rfc-editor.org/info/rfc2898>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <<http://www.rfc-editor.org/info/rfc5706>>.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, DOI 10.17487/RFC5938, August 2010, <<http://www.rfc-editor.org/info/rfc5938>>.



- [RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)", RFC 6023, DOI 10.17487/RFC6023, October 2010, <<http://www.rfc-editor.org/info/rfc6023>>.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <<http://www.rfc-editor.org/info/rfc6038>>.

Authors' Addresses

Kostas Pentikousis (editor)  
EICT GmbH  
EUREF-Campus Haus 13  
Torgauer Strasse 12-15  
10829 Berlin  
Germany

Email: [k.pentikousis@eict.de](mailto:k.pentikousis@eict.de)

Emma Zhang  
Huawei Technologies  
Huawei Building, No.3, Rd. XinXi  
Haidian District , Beijing 100095  
P. R. China

Email: [emma.zhanglijia@huawei.com](mailto:emma.zhanglijia@huawei.com)

Yang Cui  
Huawei Technologies  
Otemachi First Square 1-5-1 Otemachi  
Chiyoda-ku, Tokyo 100-0004  
Japan

Email: [cuiyang@huawei.com](mailto:cuiyang@huawei.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 10, 2015

M. Bagnulo  
UC3M  
T. Burbridge  
BT  
S. Crawford  
SamKnows  
P. Eardley  
BT  
A. Morton  
AT&T Labs  
October 7, 2014

A Reference Path and Measurement Points for Large-Scale Measurement of  
Broadband Performance  
draft-ietf-ippm-lmap-path-07

Abstract

This document defines a reference path for Large-scale Measurement of Broadband Access Performance (LMAP) and measurement points for commonly used performance metrics. Other similar measurement projects may also be able to use the extensions described here for measurement point location. The purpose is to create an efficient way to describe the location of the measurement point(s) used to conduct a particular measurement.

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 10, 2015.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Purpose and Scope . . . . .	3
3. Terms and Definitions . . . . .	4
3.1. Reference Path . . . . .	4
3.2. Subscriber . . . . .	4
3.3. Dedicated Component (Links or Nodes) . . . . .	5
3.4. Shared Component (Links or Nodes) . . . . .	5
3.5. Resource Transition Point . . . . .	5
3.6. Service Demarcation Point . . . . .	5
3.7. Managed and Un-Managed Sub-paths . . . . .	5
4. Reference Path . . . . .	6
5. Measurement Points . . . . .	7
6. Translation Between Reference Path and Various Technologies . . . . .	11
7. Example Resource Transition . . . . .	12
8. Security considerations . . . . .	13
9. IANA Considerations . . . . .	14
10. Acknowledgements . . . . .	14
11. References . . . . .	14
11.1. Normative References . . . . .	14
11.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

This document defines a reference path for Large-scale Measurement of Broadband Access Performance (LMAP) or similar measurement projects. The series of IP Performance Metrics (IPPM) RFCs have developed terms that are generally useful for path description (section 5 of [RFC2330]). There are a limited number of additional terms needing definition here, and they will be defined in this memo.

The reference path (See section 3.1 and Figure 1 of [Y.1541], including the accompanying discussion) is usually needed when attempting to communicate precisely about the components that comprise the path, often in terms of their number (hops) and geographic location. This memo takes the path definition further, by establishing a set of measurement points along the path and ascribing a unique designation to each point. This topic has been previously developed in section 5.1 of [RFC3432], and as part of the updated framework for composition and aggregation, section 4 of [RFC5835]. Section 4.1 of [RFC5835] defines the term "measurement point".

Measurement points and the paths they inhabit are often described in general terms, like "end-to-end", "user-to-user", or "access". These terms alone are insufficient for scientific method: What is an end? Where is a user located? Is the home network included?

As an illustrative example, consider a measurement agent in an LMAP system. When it reports its measurement results, rather than detailing its IP address and that of its measurement peer, it may prefer to describe the measured path segment abstractly (perhaps for privacy reasons). For instance "from a measurement agent at a home gateway to a measurement peer at a DSLAM". This memo provides the definition for such abstract 'measurement points' and therefore the portion of 'reference path' between them.

The motivation for this memo is to provide an unambiguous framework to describe measurement coverage, or scope of the reference path. This is an essential part of the meta-data to describe measurement results. Measurements conducted over different path scopes are not a valid basis for performance comparisons. We note that additional measurement context information may be necessary to support a valid comparison of results.

### 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 [RFC2119].

## 2. Purpose and Scope

The scope of this memo is to define a reference path for LMAP activities with sufficient level of detail to determine the location of different measurement points along a path without ambiguity. These conventions are likely to be useful in other measurement projects as well, and in describing the applicable measurement scope for some metrics.

The connection between the reference path and specific network technologies (with differing underlying architectures) is within the scope of this method, and examples are provided. Both wired and wireless technologies are in-scope.

The purpose is to create an efficient way to describe the location of the measurement point(s) used to conduct a particular measurement so that the measurement result will adequately described in terms of scope or coverage. This should serve many measurement uses, including:

diagnostic: where the same metric would be measured on different sub-paths bounded by measurement points (see Section 4.10 of[RFC5835]), for example to isolate the sub-path contributing the majority of impairment levels observed on a path.

comparison: where the same metric may be measured on equivalent portions of different network infrastructures, for example to compare the performance of wired and wireless home network technologies.

### 3. Terms and Definitions

This section defines key terms and concepts for the purposes of this memo.

#### 3.1. Reference Path

A reference path is a serial combination of hosts, routers, switches, links, radios, and processing elements that comprise all the network elements traversed by each packet in a flow between the source and destination hosts. A reference path also indicates the various boundaries present, such as administrative boundaries. A reference path is intended to be equally applicable to all IP and link-layer networking technologies. Therefore, the components are generically defined but their functions should have a clear counterpart or be obviously omitted in any network architecture.

#### 3.2. Subscriber

An entity (associated with one or more users) that is engaged in a subscription with a service provider. The subscriber is allowed to subscribe and un-subscribe to services, and to register a user or a list of users authorized to enjoy these services. [Q1741] Both the subscriber and service provider are allowed to set the limits relative to the use that associated users make of subscribed services.

### 3.3. Dedicated Component (Links or Nodes)

All resources of a Dedicated Component (typically a link or node on the Reference Path) are allocated to serving the traffic of an individual Subscriber. Resources include transmission time-slots, queue space, processing for encapsulation and address/port translation, and others. A Dedicated Component can affect the performance of the Reference Path, or the performance of any sub-path where the component is involved.

### 3.4. Shared Component (Links or Nodes)

A component on the Reference Path is designated a Shared Component when the traffic associated with multiple Subscribers is served by common resources.

### 3.5. Resource Transition Point

A point between Dedicated and Shared Components on a Reference Path that may be a point of significance, and is identified as a transition between two types of resources.

### 3.6. Service Demarcation Point

This is the point where service managed by the service provider begins (or ends), and varies by technology. For example, this point is usually defined as the Ethernet interface on a residential gateway or modem where the scope of a packet transfer service begins and ends. In the case of a WiFi Service, this would be an Air Interface within the intended service boundary (e.g., walls of the coffee shop). The Demarcation Point may be within an integrated endpoint using an Air Interface (e.g., Long-Term Evolution User Equipment, LTE UE). Ownership does not necessarily affect the demarcation point; a Subscriber may own all equipment on their premises, but it is likely that the service provider will certify such equipment for connection to their network, or a third-party will certify standards compliance.

### 3.7. Managed and Un-Managed Sub-paths

Service providers are responsible for the portion of the path they manage. However, most paths involve a sub-path which is beyond the management of the subscriber's service provider. This means that private networks, wireless networks using unlicensed frequencies, and the networks of other service are designated as Un-managed sub-paths. The Service Demarcation Point always divides Managed and Un-managed sub-paths.

#### 4. Reference Path

This section defines a reference path for Internet communication.

```
Subsc. -- Private -- Private -- Service-- Intra IP -- GRA -- Transit ...
device   Net #1     Net #2   Demarc.   Access   GW     GRA GW
```

```
... Transit -- GRA -- Service -- Private -- Private -- Destination
   GRA GW     GW     Demarc.   Net #n     Net #n+1   Host
```

GRA = Globally Routable Address, GW = Gateway

The following are descriptions of reference path components that may not be clear from their name alone.

- o Subsc. (Subscriber) device - This is a host that normally originates and terminates communications conducted over the IP packet transfer service.
- o Private Net #x - This is a network of devices owned and operated by the Internet Service Subscriber. In some configurations, one or more private networks and the device that provides the Service Demarcation point are collapsed in a single device (and ownership may shift to the service provider), and this should be noted as part of the path description.
- o Intra IP Access - This is the first point in the access architecture beyond the Service Demarc. where a globally routable IP address is exposed and used for routing. In architectures that use tunneling, this point may be equivalent to the Globally Routable Address Gateway (GRA GW). This point could also collapse to the device providing the Service Demarc., in principle. Only one Intra IP Access point is shown, but they can be identified in any access network.
- o GRA GW - the point of interconnection between a Service Provider's administrative domain and the rest of the Internet, where routing will depend on the GRAs in the IP header.
- o Transit GRA GW - If one or more networks intervene between the Service Provider's access networks of the Subscriber and of the Destination Host, then such networks are designated "transit" and are bounded by two Transit GRA GW.

Use of multiple IP address families in the measurement path must be noted, as the conversions between IPv4 and IPv6 certainly influence the visibility of a GRA for each family.

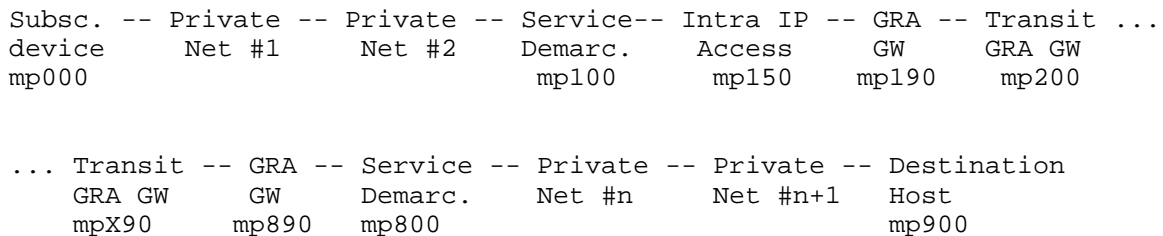
In the case that a private address space is used throughout an access architecture, then the Intra IP Access points must use the same address space as the Service Demarcation point, and the Intra IP Access points must be selected such that a test between these points produces a useful assessment of access performance (e.g., includes both shared and dedicated access link infrastructure).

## 5. Measurement Points

A key aspect of measurement points, beyond the definition in section 4.1 of [RFC5835], is that the innermost IP header and higher layer information must be accessible through some means. This is essential to measure IP metrics. There may be tunnels and/or other layers which encapsulate the innermost IP header, even adding another IP header of their own.

In general, measurement points cannot always be located exactly where desired. However, the definition in [RFC5835] and the discussion in section 5.1 of [RFC3432] indicate that allowances can be made: for example, it is nearly ideal when there are deterministic errors that can be quantified between desired and actual measurement point.

The Figure below illustrates the assignment of measurement points to selected components of the reference path.



GRA = Globally Routable Address, GW = Gateway

Figure 1

Each measurement point on a specific reference path MUST be assigned a unique number. To facilitate interpretation of the results, the measuring organisation (and whoever it shares results with) MUST have an unambiguous understanding of what path or point was measured. In order to achieve this, a set of numbering recommendations follow.



When communicating the results of measurements, the measuring organization SHOULD supply a diagram similar to Figure 1 (with the technology-specific information in examples that follow), and MUST supply it when additional measurement point numbers have been defined and used, with sufficient detail to identify measurement locations in the path.

Ideally, the consumer of measurement results would know the location of a measurement point on the reference path from the measurement point number alone, and the recommendations below provide a way to accomplish this goal. Although the initial numbering may be fully compliant with this system, network growth, consolidation, and re-arrangement, or circumstances such as ownership changes, could cause gaps in network numbers or non-monotonic measurement point number assignments along the path over time. These are examples of reasonable causes for numbering deviations which must be identified on the reference path diagram, as required above.

Whilst the numbering of a measurement point is in the context of a particular path, for simplicity the measuring organisation SHOULD use the same numbering for a device (playing the same role) on all the measurement paths through it. Similarly, whilst the measurement point numbering is in the context of a particular measuring organisation, organizations with similar technologies and architectures are encouraged to coordinate on local numbering and diagrams.

The measurement point numbering system, mpXnn, has two independent parts:

1. The X in mpXnn indicates the network number. The network with the Subscriber's device is network 0. The network of a different organization (administrative or ownership domains) SHOULD be assigned a different number. Each successive network number SHOULD be one greater than the previous network's number. Two circumstances make it necessary to designate X=9 in the Destination Host's network and X=8 for the Service Provider network at the Destination:
  - A. The number of Transit networks is unknown.
  - B. The number of Transit networks varies over time.
2. The nn in mpXnn indicates the measurement point and is locally-assigned by network X. The following conventions are suggested:

- A. 00 SHOULD be used for a measurement point at the Subscriber's device and at the Service Demarcation point or GW nearest to the Subscriber's device for Transit Networks.
- B. 90 SHOULD be used for a measurement point at the GW of a network (opposite from the Subscriber's device or Service Demarc.).
- C. In most networks, measurement point numbers SHOULD monotonically increase from the point nearest the Subscriber's device to the opposite network boundary on the path (see below).
- D. When a Destination host is part of the path, 00 SHOULD be used for a measurement point at the Destination host and at the Destination's Service Demarcation point. Measurement point numbers SHOULD monotonically increase from the point nearest the Destination's host to the opposite network boundary on the path ONLY in these networks. This directional numbering reversal allows consistent 00 designation for end hosts and Service Demarcs.
- E. 50 MAY be used for an intermediate measurement point of significance, such as a Network Address Translator (NAT).
- F. 20 MAY be used for a traffic aggregation point such as a DSLAM within a network.
- G. Any other measurement points SHOULD be assigned unused integers between 01 and 99. The assignment SHOULD be stable for at least the duration of a particular measurement study, and SHOULD avoid numbers that have been assigned to other locations within network X (unless the assignment is considered sufficiently stale). Sub-networks or domains within a network are useful locations for measurement points.

When supplying a diagram of the reference path and measurement points, the operator of the measurement system MUST indicate: the reference path, the numbers (mpXnn) of the measurement points, and the technology-specific definition of any measurement point other than X00 and X90 with sufficient detail to clearly define its location (similar to the technology-specific examples in Section 6 of this document).

If the number of intermediate networks (between the source and destination) is not known or is unstable, then this SHOULD be indicated on the diagram and results from measurement points within those networks need to be treated with caution.

## Notes:

- o The terminology "on-net" and "off-net" is sometimes used when referring to the Subscriber's Internet Service Provider (ISP) measurement coverage. With respect to the reference path, tests between mp100 and mp190 are "on-net".
- o Widely deployed broadband Internet access measurements have used pass-through devices[SK] (at the subscriber's location) directly connected to the service demarcation point: this would be located at mp100.
- o The networking technology must be indicated for the measurement points used, especially the interface standard and configured speed (because the measurement connectivity itself can be a limiting factor for the results).
- o If it can be shown that a link connecting to a measurement point has reliably deterministic performance or negligible impairments, then the remote end of the connecting link is an equivalent point for some methods of measurement (although those methods should describe this possibility in detail; it is not in-scope to provide such methods here). In any case, the presence of a link and claimed equivalent measurement point must be reported.
- o Some access network architectures may have an additional traffic aggregation device between mp100 and mp150. Use of a measurement point at this location would require a local number and diagram.
- o A Carrier Grade NAT (CGN) deployed in the Service Provider's access network would be positioned between mp100 and mp190, and the egress side of the CGN may be designated mp150. mp150 is generally an intermediate measurement point in the same address space as mp190.
- o In the case that private address space is used in an access architecture, then mp100 may need to use the same address space as its "on-net" measurement point counterpart, so that a test between these points produces a useful assessment of network performance. Tests between mp000 and mp100 could use a different private address space, and when the globally-routable side of a CGN is at mp150, then the private address side of the CGN could be designated mp149 for tests with mp100.
- o Measurement points at Transit GRA GWs are numbered mpX00 and mpX90, where X is the lowest positive integer not already used in the path. The GW of the first transit network is shown, with point mp200 and the last transit network GW with mpX90.

## 6. Translation Between Reference Path and Various Technologies

This section and those that follow are intended to provide example mappings between particular network technologies and the reference path.

We provide an example for 3G Cellular access below.

Subscriber	--	Private	---	Service	-----	GRA	---	Transit	...
device		Net #1		Demarc.		GW		GRA GW	
mp000				mp100		mp190		mp200	

	_____UE_____		_____RAN+Core_____		_____GGSN_____	
	_____Un-managed sub-path_____		_____Managed sub-path_____			

GRA = Globally Routable Address, GW = Gateway, UE = User Equipment,  
RAN = Radio Access Network, GGSN = Gateway GPRS Support Node.

We next provide an example of DSL access. Consider the case where:

- o The Customer Premises Equipment (CPE) has a NAT device that is configured with a public IP address.
- o The CPE is a home router that has also incorporated a WiFi access point and this is the only networking device in the home network, all endpoints attach directly to the CPE through the WiFi access.

We believe this is a fairly common configuration in some parts of the world and fairly simple as well.

This case would map into the defined reference measurement points as follows:

Subsc.	--	Private	--	Private	--	Service	--	Intra IP	--	GRA	--	Transit	...
device		Net #1		Net #2		Demarc.		Access		GW		GRA GW	
mp000						mp100		mp150		mp190		mp200	

	--UE--		-----CPE/NAT-----		-----		-BRAS-		-----	
					-----DSL Network---					
	_____Un-managed sub-path_____		_____Managed sub-path_____							

GRA = Globally Routable Address, GW = Gateway, BRAS = Broadband Remote Access Server

Consider next another access network case where:

- o The Customer Premises Equipment (CPE) is a NAT device that is configured with a private IP address.
- o There is a Carrier Grade NAT (CGN) located deep in the Access ISP network.
- o The CPE is a home router that has also incorporated a WiFi access point and this is the only networking device in the home network, all endpoints attach directly to the CPE through the WiFi access.

We believe this is becoming a fairly common configuration in some parts of the world.

This case would map into the defined reference measurement points as follows:

Subsc. device	-- Private Net #1	-----	Service-- Demarc.	Intra IP -- Access	GRA -- GW	Transit ... GRA GW
mp000			mp100	mp150	mp190	mp200
--UE--	-----CPE/NAT-----		-----	-CGN-	-----	
	Un-managed sub-path		--Access Network--			
			_Managed sub-path_			

GRA = Globally Routable Address, GW = Gateway

## 7. Example Resource Transition

This section gives an example of Shared and Dedicated portions with the reference path. This example shows two Resource Transition Points.

Consider the case where:

- o The CPE consists of a wired Residential GW and modem (Private Net#2) connected to a WiFi access point (Private Net#1). The Subscriber device (UE) attaches to the CPE through the WiFi access.
- o The WiFi subnetwork (Private Net#1) shares unlicensed radio channel resources with other WiFi access networks (and potentially other sources of interference), thus this is a Shared portion of the path.
- o The wired subnetwork (Private Net#2) and a portion of the Service Provider's Network are Dedicated Resources (for a single Subscriber), thus there is a Resource Transition Point between (Private Net#1) and (Private Net#2).

- o Subscriber traffic shares common resources with other subscribers upon reaching the Carrier Grade NAT (CGN), thus there is a Resource Transition Point and further network components are designated as Shared Resources.

We believe this is a fairly common configuration in parts of the world.

This case would map into the defined reference measurement points as follows:

```

Subsc. -- Private -- Private -- Access -- Intra IP -- GRA -- Transit ...
device      Net #1      Net #2      Demarc.      Access      GW      GRA GW
mp000
|--UE--|-----CPE/NAT-----|-----| -CGN- |-----|
      |   WiFi   | 1000Base-T | --Access Network--|
      |
      | -Shared--|RT|-----Dedicated-----| RT |-----Shared-----...
      |_____Un-managed sub-path_____||_Managed sub-path_|

```

GRA = Globally Routable Address, GW = Gateway, RT = Resource Transition Point

## 8. Security considerations

Specification of a Reference Path and identification of measurement points on the path represent agreements among interested parties, and they present no threat to the implementors of this memo, or to the Internet resulting from implementation of the guidelines provided here.

Attacks at end hosts or identified measurement points are possible. However, there is no requirement to include IP addresses of hosts or other network devices in a reference path with measurement points that is compliant with this memo. As a result, the path diagrams with measurement point designation numbers do not aid such attacks.

Most network operators' diagrams of reference paths will bear a close resemblance to similar diagrams in relevant standards or other publicly available documents. However, when an operator must include atypical network details in their diagram, e.g., to explain why a longer latency measurement is expected, then the diagram reveals some topological details and should be marked as confidential and shared with others under a specific agreement.

When considering privacy of those involved in measurement or those whose traffic is measured, there may be sensitive information

communicated to recipients of the network diagrams illustrating paths and measurement points described above. We refer the reader to the privacy considerations described in the Large Scale Measurement of Broadband Performance (LMAP) Framework [I-D.ietf-lmap-framework], which covers active and passive measurement techniques and supporting material on measurement context. For example, the value of sensitive information can be further diluted by summarising measurement results over many individuals or areas served by the provider. There is an opportunity enabled by forming anonymity sets described in [RFC6973] based on the reference path and measurement points in this memo. For example, all measurements from the Subscriber device can be identified as "mp000", instead of using the IP address or other device information. The same anonymisation applies to the Internet Service Provider, where their Internet gateway would be referred to as "mpl90".

## 9. IANA Considerations

This memo makes no requests for IANA consideration.

## 10. Acknowledgements

Thanks to Matt Mathis, Charles Cook, Dan Romascanu, Lingli Deng, and Spencer Dawkins for review and comments.

## 11. References

### 11.1. Normative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.

### 11.2. Informative References

- [I-D.ietf-lmap-framework]  
Eardley, P., Morton, A., Bagnulo, M., Burbridge, T.,  
Aitken, P., and A. Akhter, "A framework for large-scale  
measurement platforms (LMAP)", draft-ietf-lmap-  
framework-08 (work in progress), August 2014.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J.,  
Morris, J., Hansen, M., and R. Smith, "Privacy  
Considerations for Internet Protocols", RFC 6973, July  
2013.
- [SK] Crawford, Sam., "Test Methodology White Paper", SamKnows  
Whitebox Briefing Note  
<http://www.samknows.com/broadband/index.php>, July 2011.
- [Q1741] Q.1741.7, , "IMT-2000 references to Release 9 of GSM-  
evolved UMTS core network",  
<http://www.itu.int/rec/T-REC-Q.1741.7/en>, November 2011.
- [Y.1541] Y.1541, , "Network performance objectives for IP-based  
services", <http://www.itu.int/rec/T-REC-Y.1541/en>,  
November 2011.

## Authors' Addresses

Marcelo Bagnulo  
Universidad Carlos III de Madrid  
Av. Universidad 30  
Leganes, Madrid 28911  
SPAIN

Phone: 34 91 6249500  
Email: [marcelo@it.uc3m.es](mailto:marcelo@it.uc3m.es)  
URI: <http://www.it.uc3m.es>

Trevor Burbridge  
BT  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: [trevor.burbridge@bt.com](mailto:trevor.burbridge@bt.com)



Sam Crawford  
SamKnows

Email: sam@samknows.com

Phil Eardley  
BT  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: philip.eardley@bt.com

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

Email: acmorton@att.com

IP Performance Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: March 19, 2018

M. Mathis  
Google, Inc  
A. Morton  
AT&T Labs  
September 15, 2017

Model Based Metrics for Bulk Transport Capacity  
draft-ietf-ippm-model-based-metrics-13.txt

Abstract

We introduce a new class of Model Based Metrics designed to assess if a complete Internet path can be expected to meet a predefined Target Transport Performance by applying a suite of IP diagnostic tests to successive subpaths. The subpath-at-a-time tests can be robustly applied to critical infrastructure, such as network interconnections or even individual devices, to accurately detect if any part of the infrastructure will prevent paths traversing it from meeting the Target Transport Performance.

Model Based Metrics rely on mathematical models to specify a Targeted Suite of IP Diagnostic tests, designed to assess whether common transport protocols can be expected to meet a predetermined Target Transport Performance over an Internet path.

For Bulk Transport Capacity the IP diagnostics are built using test streams and statistical criteria for evaluating the packet transfer that mimic TCP over the complete path. The temporal structure of the test stream (bursts, etc) mimic TCP or other transport protocol carrying bulk data over a long path. However they are constructed to be independent of the details of the subpath under test, end systems or applications. Likewise the success criteria evaluates the packet transfer statistics of the subpath against criteria determined by protocol performance models applied to the Target Transport Performance of the complete path. The success criteria also does not depend on the details of the subpath, end systems or application.

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 March 19, 2018.

#### Copyright Notice

Copyright (c) 2017 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. Version Control . . . . .	5
2. Overview . . . . .	8
3. Terminology . . . . .	10
4. Background . . . . .	17
4.1. TCP properties . . . . .	18
4.2. Diagnostic Approach . . . . .	20
4.3. New requirements relative to RFC 2330 . . . . .	21
5. Common Models and Parameters . . . . .	22
5.1. Target End-to-end parameters . . . . .	22
5.2. Common Model Calculations . . . . .	23
5.3. Parameter Derating . . . . .	24
5.4. Test Preconditions . . . . .	24
6. Generating test streams . . . . .	25
6.1. Mimicking slowstart . . . . .	26
6.2. Constant window pseudo CBR . . . . .	27
6.3. Scanned window pseudo CBR . . . . .	28
6.4. Concurrent or channelized testing . . . . .	29
7. Interpreting the Results . . . . .	30
7.1. Test outcomes . . . . .	30
7.2. Statistical criteria for estimating run_length . . . . .	31
7.3. Reordering Tolerance . . . . .	34
8. IP Diagnostic Tests . . . . .	34
8.1. Basic Data Rate and Packet Transfer Tests . . . . .	35

8.1.1.	Delivery Statistics at Paced Full Data Rate . . . . .	35
8.1.2.	Delivery Statistics at Full Data Windowed Rate . . . . .	35
8.1.3.	Background Packet Transfer Statistics Tests . . . . .	35
8.2.	Standing Queue Tests . . . . .	36
8.2.1.	Congestion Avoidance . . . . .	37
8.2.2.	Bufferbloat . . . . .	37
8.2.3.	Non excessive loss . . . . .	38
8.2.4.	Duplex Self Interference . . . . .	38
8.3.	Slowstart tests . . . . .	39
8.3.1.	Full Window slowstart test . . . . .	39
8.3.2.	Slowstart AQM test . . . . .	39
8.4.	Sender Rate Burst tests . . . . .	40
8.5.	Combined and Implicit Tests . . . . .	41
8.5.1.	Sustained Bursts Test . . . . .	41
8.5.2.	Passive Measurements . . . . .	42
9.	An Example . . . . .	43
9.1.	Observations about applicability . . . . .	44
10.	Validation . . . . .	44
11.	Security Considerations . . . . .	46
12.	Acknowledgments . . . . .	46
13.	IANA Considerations . . . . .	47
14.	Informative References . . . . .	47
Appendix A.	Model Derivations . . . . .	51
A.1.	Queueless Reno . . . . .	51
Appendix B.	The effects of ACK scheduling . . . . .	52
Appendix C.	Version Control . . . . .	53
Authors' Addresses	. . . . .	53

## 1. Introduction

Model Based Metrics (MBM) rely on peer-reviewed mathematical models to specify a Targeted Suite of IP Diagnostic tests, designed to assess whether common transport protocols can be expected to meet a predetermined Target Transport Performance over an Internet path. This note describes the modeling framework to derive the test parameters for assessing an Internet path's ability to support a predetermined Bulk Transport Capacity.

Each test in the Targeted IP Diagnostic Suite (TIDS) measures some aspect of IP packet transfer needed to meet the Target Transport Performance. For Bulk Transport Capacity the TIDS includes IP diagnostic tests to verify that there is: sufficient IP capacity (data rate); sufficient queue space at bottlenecks to absorb and deliver typical transport bursts; and that the background packet loss ratio is low enough not to interfere with congestion control; and other properties described below. Unlike typical IPPM metrics which yield measures of network properties, Model Based Metrics nominally yield pass/fail evaluations of the ability of standard transport

protocols to meet the specific performance objective over some network path.

In most cases, the IP diagnostic tests can be implemented by combining existing IPPM metrics with additional controls for generating test streams having a specified temporal structure (bursts or standing queues caused by constant bit rate streams, etc.) and statistical criteria for evaluating packet transfer. The temporal structure of the test streams mimic transport protocol behavior over the complete path; the statistical criteria models the transport protocol's response to less than ideal IP packet transfer. In control theory terms, the tests are "open loop". Note that running a test requires the coordinated activity of sending and receiving measurement points.

This note addresses Bulk Transport Capacity. It describes an alternative to the approach presented in "A Framework for Defining Empirical Bulk Transfer Capacity Metrics" [RFC3148]. Other Model Based Metrics may cover other applications and transports, such as VoIP over UDP and RTP, and new transport protocols.

This note assumes a traditional Reno TCP style self clocked, window controlled transport protocol that uses packet loss and ECN CE marks for congestion feedback. There are currently some experimental protocols and congestion control algorithms that are rate based or otherwise fall outside of these assumptions. In the future these new protocols and algorithms may call for revised models.

The MBM approach, mapping Target Transport Performance to a Targeted IP Diagnostic Suite (TIDS) of IP tests, solves some intrinsic problems with using TCP or other throughput maximizing protocols for measurement. In particular all throughput maximizing protocols (and TCP congestion control in particular) cause some level of congestion in order to detect when they have reached the available capacity limitation of the network. This self inflicted congestion obscures the network properties of interest and introduces non-linear dynamic equilibrium behaviors that make any resulting measurements useless as metrics because they have no predictive value for conditions or paths different than that of the measurement itself. In order to prevent these effects it is necessary to avoid the effects of TCP congestion control in the measurement method. These issues are discussed at length in Section 4. Readers whom are unfamiliar with basic properties of TCP and TCP-like congestion control may find it easier to start at Section 4 or Section 4.1.

A Targeted IP Diagnostic Suite does not have such difficulties. IP diagnostics can be constructed such that they make strong statistical statements about path properties that are independent of the

measurement details, such as vantage and choice of measurement points.

### 1.1. Version Control

RFC Editor: Please remove this entire subsection prior to publication.

REF Editor: The reference to draft-ietf-tcpm-rack is to attribute an idea. This document should not block waiting for the completion of that one.

Please send comments about this draft to [ippm@ietf.org](mailto:ippm@ietf.org). See <http://goo.gl/02tkD> for more information including: interim drafts, an up to date todo list and information on contributing.

Formatted: Fri Sep 15 15:07:50 PDT 2017

Changes since -11 draft:

- o (From IESG review comments.)
- o Ben Campbell: Shorten the Abstract.
- o Mirja Kuhlewind: Reduced redundancy. (See message)
- o MK: Mention open loop in the introduction.
- o MK: Spelled out ECN and reference RFC3168.
- o MK: Added a paragraph to the introduction about assuming a traditional self clocked, window controlled transport protocol.
- o MK: Added language about initial window to the list at about bursts at the end of section 4.1.
- o MK: Network power is defined in the terminology section.
- o MK: The introduction mention coordinated activity of both endpoints.
- o MK: The security section restates that some of the tests are not intended for frequent monitoring tests as the high load can impact other traffic negatively.
- o MK: Restored "Informative References" section name.
- o And a few minor nits.

Changes since -10 draft:

- o A few more nits from various sources.
- o (From IETF LC review comments.)
- o David Mandelberg: design metrics to prevent DDOS.
- o From Robert Sparks:
  - \* Remove all legacy 2119 language.
  - \* Fixed Xr notation inconsistency.
  - \* Adjusted abstract: tests are only partially specified.

- \* Avoid rather than suppress the effects of congestion control
- \* Removed the unnecessary, excessively abstract and unclear thought about IP vs TCP measurements.
- \* Changed "thwarted" to "not fulfilled".
- \* Qualified language about burst models.
- \* Replaced "infinitesimal" with other language.
- \* Added citations for the reordering strawman.
- \* Pointed out that pseudo CBR tests depend on self clock.
- \* Fixed some run on sentences.
- o Update language to reflect RFC7567, AQM recommendations.
- o Suggestion from Merry Mou (MIT)

Changes since -09 draft:

- o Five last minute editing nits.

Changes since -08 draft:

- o Language, spelling and usage nits.
- o Expanded the abstract describe the models.
- o Remove superfluous standards like language
- o Remove superfluous "future technology" language.
- o Interconnects -> network interconnections.
- o Added more labels to Figure 1.
- o Defined Bulk Transport.
- o Clarified "implied bottleneck IP capacity"
- o Clarified the history of the BTC metrics.
- o Clarified stochastic vs non-stochastic test traffic generation.
- o Reworked Fig 2 and 6.1 "Mimicking slowstart"
- o Described the unsynchronized parallel stream failure case.
- o Discussed how to measure devices that use virtual queues.
- o Changed section 8.5.2 (Streaming Media) to be Passive Measurements.

Changes since -07 draft:

- o Sharpened the use of "statistical criteria"
- o Sharpened the definition of test\_window, and removed related redundant text in several places
- o Clarified "equilibrium" as "dynamic equilibrium, similar to processes observed in chemistry"
- o Properly explained "Heisenberg" as "observer effect"
- o Added the observation from RFC 6576 that HW and SW congestion control implementations do not generally give the same results.
- o Noted that IP and application metrics differ as to how overhead is handled. MBM is explicit about how it handles overhead.
- o Clarified the language and added a new reference about the problems caused by token bucket policers.

- o Added an subsection in the example that comments on some of issues that need to be mentioned in a future usage or applicability doc.
- o Updated ippm-2680-bis to RFC7680
- o Many terminology, punctuation and spelling nits.

Changes since -06 draft:

- o More language nits:
  - \* "Targeted IP Diagnostic Suite (TIDS)" replaces "Targeted Diagnostic Suite (TDS)".
  - \* "implied bottleneck IP capacity" replaces "implied bottleneck IP rate".
  - \* Updated to ECN CE Marks.
  - \* Added "specified temporal structure"
  - \* "test stream" replaces "test traffic"
  - \* "packet transfer" replaces "packet delivery"
  - \* Reworked discussion of slowstart, bursts and pacing.
  - \* RFC 7567 replaces RFC 2309.

Changes since -05 draft:

- o Wordsmithing on sections overhauled in -05 draft.
- o Reorganized the document:
  - \* Relocated subsection "Preconditions".
  - \* Relocated subsection "New Requirements relative to RFC 2330".
- o Addressed nits and not so nits by Ruediger Geib. (Thanks!)
- o Substantially tightened the entire definitions section.
- o Many terminology changes, to better conform to other docs :
  - \* IP rate and IP capacity (following RFC 5136) replaces various forms of link data rate.
  - \* subpath replaces link.
  - \* target\_window\_size replaces target\_pipe\_size.
  - \* implied bottleneck IP rate replaces effective bottleneck link rate.
  - \* Packet delivery statistics replaces delivery statistics.

Changes since -04 draft:

- o The introduction was heavily overhauled: split into a separate introduction and overview.
- o The new shorter introduction:
  - \* Is a problem statement;
  - \* This document provides a framework;
  - \* That it replaces TCP measurement by IP tests;



- \* That the results are pass/fail.
- o Added a diagram of the framework to the overview
- o and introduces all of the elements of the framework.
- o Renumbered sections, reducing the depth of some section numbers.
- o Updated definitions to better agree with other documents:
- \* Reordered section 2
- \* Bulk [data] performance -> Bulk Transport Capacity, everywhere including the title.
- \* loss rate and loss probability -> packet loss ratio
- \* end-to-end path -> complete path
- \* [end-to-end][target] performance -> Target Transport Performance
- \* load test -> capacity test

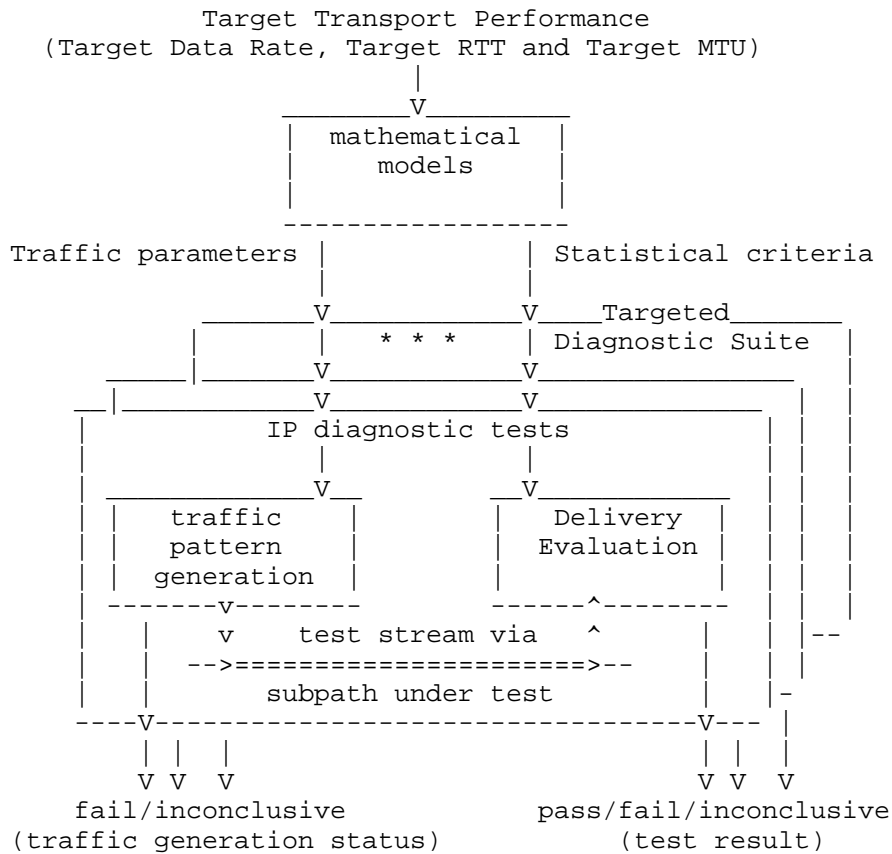
## 2. Overview

This document describes a modeling framework for deriving a Targeted IP Diagnostic Suite from a predetermined Target Transport Performance. It is not a complete specification, and relies on other standards documents to define important details such as packet Type-P selection, sampling techniques, vantage selection, etc. We imagine Fully Specified - Targeted IP Diagnostic Suites (FS-TIDS), that define all of these details. We use Targeted IP Diagnostic Suite (TIDS) to refer to the subset of such a specification that is in scope for this document. This terminology is defined in Section 3.

Section 4 describes some key aspects of TCP behavior and what they imply about the requirements for IP packet transfer. Most of the IP diagnostic tests needed to confirm that the path meets these properties can be built on existing IPPM metrics, with the addition of statistical criteria for evaluating packet transfer and in a few cases, new mechanisms to implement the required temporal structure. (One group of tests, the standing queue tests described in Section 8.2, don't correspond to existing IPPM metrics, but suitable new IPPM metrics can be patterned after the existing definitions.)

Figure 1 shows the MBM modeling and measurement framework. The Target Transport Performance, at the top of the figure, is determined by the needs of the user or application, outside the scope of this document. For Bulk Transport Capacity, the main performance parameter of interest is the Target Data Rate. However, since TCP's ability to compensate for less than ideal network conditions is fundamentally affected by the Round Trip Time (RTT) and the Maximum Transmission Unit (MTU) of the complete path, these parameters must also be specified in advance based on knowledge about the intended application setting. They may reflect a specific application over a real path through the Internet or an idealized application and

hypothetical path representing a typical user community. Section 5 describes the common parameters and models derived from the Target Transport Performance.



Overall Modeling Framework

Figure 1

Mathematical TCP models are used to determine Traffic parameters and subsequently to design traffic patterns that mimic TCP or other transport protocol delivering bulk data and operating at the Target Data Rate, MTU and RTT over a full range of conditions, including flows that are bursty at multiple time scales. The traffic patterns are generated based on the three Target parameters of complete path and independent of the properties of individual subpaths using the techniques described in Section 6. As much as possible the test streams are generated deterministically (precomputed) to minimize the extent to which test methodology, measurement points, measurement

vantage or path partitioning affect the details of the measurement traffic.

Section 7 describes packet transfer statistics and methods to test them against the statistical criteria provided by the mathematical models. Since the statistical criteria typically apply to the complete path (a composition of subpaths) [RFC6049], in situ testing requires that the end-to-end statistical criteria be apportioned as separate criteria for each subpath. Subpaths that are expected to be bottlenecks would then be permitted to contribute a larger fraction of the end-to-end packet loss budget. In compensation, subpaths that are not expected to exhibit bottlenecks must be constrained to contribute less packet loss. Thus the statistical criteria for each subpath in each test of a TIDS is an apportioned share of the end-to-end statistical criteria for the complete path which was determined by the mathematical model.

Section 8 describes the suite of individual tests needed to verify all of required IP delivery properties. A subpath passes if and only if all of the individual IP diagnostic tests pass. Any subpath that fails any test indicates that some users are likely to fail to attain their Target Transport Performance under some conditions. In addition to passing or failing, a test can be deemed to be inconclusive for a number of reasons including: the precomputed traffic pattern was not accurately generated; the measurement results were not statistically significant; and others such as failing to meet some required test preconditions. If all tests pass but some are inconclusive, then the entire suite is deemed to be inconclusive.

In Section 9 we present an example TIDS that might be representative of High Definition (HD) video, and illustrate how Model Based Metrics can be used to address difficult measurement situations, such as confirming that inter-carrier exchanges have sufficient performance and capacity to deliver HD video between ISPs.

Since there is some uncertainty in the modeling process, Section 10 describes a validation procedure to diagnose and minimize false positive and false negative results.

### 3. Terminology

Terms containing underscores (rather than spaces) appear in equations and typically have algorithmic definitions.

General Terminology:

Target: A general term for any parameter specified by or derived from the user's application or transport performance requirements.

**Target Transport Performance:** Application or transport performance target values for the complete path. For Bulk Transport Capacity defined in this note the Target Transport Performance includes the Target Data Rate, Target RTT and Target MTU as described below.

**Target Data Rate:** The specified application data rate required for an application's proper operation. Conventional Bulk Transport Capacity (BTC) metrics are focused on the Target Data Rate, however these metrics had little or no predictive value because they do not consider the effects of the other two parameters of the Target Transport Performance, the RTT and MTU of the complete paths.

**Target RTT (Round Trip Time):** The specified baseline (minimum) RTT of the longest complete path over which the user expects to be able to meet the target performance. TCP and other transport protocol's ability to compensate for path problems is generally proportional to the number of round trips per second. The Target RTT determines both key parameters of the traffic patterns (e.g. burst sizes) and the thresholds on acceptable IP packet transfer statistics. The Target RTT must be specified considering appropriate packets sizes: MTU sized packets on the forward path, ACK sized packets (typically header\_overhead) on the return path. Note that Target RTT is specified and not measured, MBM measurements derived for a given target\_RTT will be applicable to any path with a smaller RTTs.

**Target MTU (Maximum Transmission Unit):** The specified maximum MTU supported by the complete path the over which the application expects to meet the target performance. In this document assume a 1500 Byte MTU unless otherwise specified. If some subpath has a smaller MTU, then it becomes the Target MTU for the complete path, and all model calculations and subpath tests must use the same smaller MTU.

**Targeted IP Diagnostic Suite (TIDS):** A set of IP diagnostic tests designed to determine if an otherwise ideal complete path containing the subpath under test can sustain flows at a specific target\_data\_rate using target\_MTU sized packets when the RTT of the complete path is target\_RTT.

**Fully Specified Targeted IP Diagnostic Suite (FS-TIDS):** A TIDS together with additional specification such as measurement packet type ("type-p" [RFC2330]), etc. which are out of scope for this document, but need to be drawn from other standards documents.

**Bulk Transport Capacity:** Bulk Transport Capacity Metrics evaluate an Internet path's ability to carry bulk data, such as large files, streaming (non-real time) video, and under some conditions, web images and other content. Prior efforts to define BTC metrics have been based on [RFC3148], which predates our understanding of TCP and the requirements described in Section 4. In general "Bulk Transport" indicates that performance is determined by the interplay between the network, cross traffic and congestion

control in the transport protocol. It excludes situations where performance is dominated by the RTT alone (e.g. transactions) or bottlenecks elsewhere, such as in the application itself.

IP diagnostic tests: Measurements or diagnostics to determine if packet transfer statistics meet some precomputed target.

traffic patterns: The temporal patterns or burstiness of traffic generated by applications over transport protocols such as TCP. There are several mechanisms that cause bursts at various time scales as described in Section 4.1. Our goal here is to mimic the range of common patterns (burst sizes and rates, etc), without tying our applicability to specific applications, implementations or technologies, which are sure to become stale.

Explicit Congestion Notification (ECN): See [RFC3168].

packet transfer statistics: Raw, detailed or summary statistics about packet transfer properties of the IP layer including packet losses, ECN Congestion Experienced (CE) marks, reordering, or any other properties that may be germane to transport performance.

packet loss ratio: As defined in [RFC7680].

apportioned: To divide and allocate, for example budgeting packet loss across multiple subpaths such that the losses will accumulate to less than a specified end-to-end loss ratio. Apportioning metrics is essentially the inverse of the process described in [RFC5835].

open loop: A control theory term used to describe a class of techniques where systems that naturally exhibit circular dependencies can be analyzed by suppressing some of the dependencies, such that the resulting dependency graph is acyclic.

Terminology about paths, etc. See [RFC2330] and [RFC7398] for existing terms and definitions.

data sender: Host sending data and receiving ACKs.

data receiver: Host receiving data and sending ACKs.

complete path: The end-to-end path from the data sender to the data receiver.

subpath: A portion of the complete path. Note that there is no requirement that subpaths be non-overlapping. A subpath can be as small as a single device, link or interface.

measurement point: Measurement points as described in [RFC7398].

test path: A path between two measurement points that includes a subpath of the complete path under test. If the measurement points are off path, the test path may include "test leads" between the measurement points and the subpath.

dominant bottleneck: The bottleneck that generally determines most of packet transfer statistics for the entire path. It typically determines a flow's self clock timing, packet loss and ECN Congestion Experienced (CE) marking rate, with other potential

bottlenecks having less effect on the packet transfer statistics.  
See Section 4.1 on TCP properties.  
front path: The subpath from the data sender to the dominant bottleneck.  
back path: The subpath from the dominant bottleneck to the receiver.  
return path: The path taken by the ACKs from the data receiver to the data sender.  
cross traffic: Other, potentially interfering, traffic competing for network resources (bandwidth and/or queue capacity).

Properties determined by the complete path and application. These are described in more detail in Section 5.1.

Application Data Rate: General term for the data rate as seen by the application above the transport layer in bytes per second. This is the payload data rate, and explicitly excludes transport and lower level headers (TCP/IP or other protocols), retransmissions and other overhead that is not part to the total quantity of data delivered to the application.

IP rate: The actual number of IP-layer bytes delivered through a subpath, per unit time, including TCP and IP headers, retransmits and other TCP/IP overhead. Follows from IP-type-P Link Usage [RFC5136].

IP capacity: The maximum number of IP-layer bytes that can be transmitted through a subpath, per unit time, including TCP and IP headers, retransmits and other TCP/IP overhead. Follows from IP-type-P Link Capacity [RFC5136].

bottleneck IP capacity: The IP capacity of the dominant bottleneck in the forward path. All throughput maximizing protocols estimate this capacity by observing the IP rate delivered through the bottleneck. Most protocols derive their self clocks from the timing of this data. See Section 4.1 and Appendix B for more details.

implied bottleneck IP capacity: This is the bottleneck IP capacity implied by the ACKs returning from the receiver. It is determined by looking at how much application data the ACK stream at the sender reports delivered to the data receiver per unit time at various time scales. If the return path is thinning, batching or otherwise altering the ACK timing the implied bottleneck IP capacity over short time scales might be substantially larger than the bottleneck IP capacity averaged over a full RTT. Since TCP derives its clock from the data delivered through the bottleneck, the front path must have sufficient buffering to absorb any data bursts at the dimensions (size and IP rate) implied by the ACK stream, which are potentially doubled during slowstart. If the return path is not altering the ACK stream, then the implied bottleneck IP capacity will be the same as the bottleneck IP capacity. See Section 4.1 and Appendix B for more details.

sender interface rate: The IP rate which corresponds to the IP capacity of the data sender's interface. Due to sender efficiency algorithms including technologies such as TCP segmentation offload (TSO), nearly all modern servers deliver data in bursts at full interface link rate. Today 1 or 10 Gb/s are typical.

Header\_overhead: The IP and TCP header sizes, which are the portion of each MTU not available for carrying application payload. Without loss of generality this is assumed to be the size for returning acknowledgments (ACKs). For TCP, the Maximum Segment Size (MSS) is the Target MTU minus the header\_overhead.

Basic parameters common to models and subpath tests are defined here are described in more detail in Section 5.2. Note that these are mixed between application transport performance (excludes headers) and IP performance (which include TCP headers and retransmissions as part of the IP payload).

Network power: The observed data rate divided by the observed RTT. Network power indicates how effectively a transport protocol is filling a network.

Window [size]: The total quantity of data carried by packets in-flight plus the data represented by ACKs circulating in the network is referred to as the window. See Section 4.1. Sometimes used with other qualifiers (congestion window, cwnd or receiver window) to indicate which mechanism is controlling the window.

pipe size: A general term for number of packets needed in flight (the window size) to exactly fill some network path or subpath. It corresponds to the window size which maximizes network power. Often used with additional qualifiers to specify which path, or under what conditions, etc.

target\_window\_size: The average number of packets in flight (the window size) needed to meet the Target Data Rate, for the specified Target RTT, and MTU. It implies the scale of the bursts that the network might experience.

run length: A general term for the observed, measured, or specified number of packets that are (expected to be) delivered between losses or ECN Congestion Experienced (CE) marks. Nominally one over the sum of the loss and ECN CE marking probabilities, if there are independently and identically distributed.

target\_run\_length: The target\_run\_length is an estimate of the minimum number of non-congestion marked packets needed between losses or ECN Congestion Experienced (CE) marks necessary to attain the target\_data\_rate over a path with the specified target\_RTT and target\_MTU, as computed by a mathematical model of TCP congestion control. A reference calculation is shown in Section 5.2 and alternatives in Appendix A

reference target\_run\_length: target\_run\_length computed precisely by the method in Section 5.2. This is likely to be slightly more conservative than required by modern TCP implementations.

Ancillary parameters used for some tests:

derating: Under some conditions the standard models are too conservative. The modeling framework permits some latitude in relaxing or "derating" some test parameters as described in Section 5.3 in exchange for a more stringent TIDS validation procedures, described in Section 10. Models can be derated by including a multiplicative derating factor to make tests less stringent.

subpath\_IP\_capacity: The IP capacity of a specific subpath.

test path: A subpath of a complete path under test.

test\_path\_RTT: The RTT observed between two measurement points using packet sizes that are consistent with the transport protocol. This is generally MTU sized packets of the forward path, header\_overhead sized packets on the return path.

test\_path\_pipe: The pipe size of a test path. Nominally the test\_path\_RTT times the test path IP\_capacity.

test\_window: The smallest window sufficient to meet or exceed the target\_rate when operating with a pure self clock over a test path. The test\_window is typically given by  $\text{ceiling}(\text{target\_data\_rate} * \text{test\_path\_RTT} / (\text{target\_MTU} - \text{header\_overhead}))$  but see the discussion in Appendix B about the effects of channel scheduling on RTT. On some test paths the test\_window may need to be adjusted slightly to compensate for the RTT being inflated by the devices that schedule packets.

The terminology below is used to define temporal patterns for test stream. These patterns are designed to mimic TCP behavior, as described in Section 4.1.

packet headway: Time interval between packets, specified from the start of one to the start of the next. e.g. If packets are sent with a 1 mS headway, there will be exactly 1000 packets per second.

burst headway: Time interval between bursts, specified from the start of the first packet one burst to the start of the first packet of the next burst. e.g. If 4 packet bursts are sent with a 1 mS burst headway, there will be exactly 4000 packets per second.

paced single packets: Send individual packets at the specified rate or packet headway.

paced bursts: Send bursts on a timer. Specify any 3 of: average data rate, packet size, burst size (number of packets) and burst headway (burst start to start). By default the bursts are assumed to occur at full sender interface rate, such that the packet



headway within each burst is the minimum supported by the sender's interface. Under some conditions it is useful to explicitly specify the packet headway within each burst.

slowstart rate: Mimic TCP slowstart by sending 4 packet paced bursts at an average data rate equal to twice the implied bottleneck IP capacity (but not more than the sender interface rate). This is a two level burst pattern described in more detail in Section 6.1. If the implied bottleneck IP capacity is more than half of the sender interface rate, slowstart rate becomes sender interface rate.

slowstart burst: Mimic one round of TCP slowstart by sending a specified number of packets in a two level burst pattern that resembles slowstart.

repeated slowstart bursts: Repeat Slowstart bursts once per target\_RTT. For TCP each burst would be twice as large as the prior burst, and the sequence would end at the first ECN CE mark or lost packet. For measurement, all slowstart bursts would be the same size (nominally target\_window\_size but other sizes might be specified), and the ECN CE marks and lost packets are counted.

The tests described in this note can be grouped according to their applicability.

Capacity tests: Capacity tests determine if a network subpath has sufficient capacity to deliver the Target Transport Performance. As long as the test stream is within the proper envelope for the Target Transport Performance, the average packet losses or ECN Congestion Experienced (CE) marks must be below the statistical criteria computed by the model. As such, capacity tests reflect parameters that can transition from passing to failing as a consequence of cross traffic, additional presented load or the actions of other network users. By definition, capacity tests also consume significant network resources (data capacity and/or queue buffer space), and the test schedules must be balanced by their cost.

Monitoring tests: Monitoring tests are designed to capture the most important aspects of a capacity test, but without presenting excessive ongoing load themselves. As such they may miss some details of the network's performance, but can serve as a useful reduced-cost proxy for a capacity test, for example to support continuous production network monitoring.

Engineering tests: Engineering tests evaluate how network algorithms (such as AQM and channel allocation) interact with TCP-style self clocked protocols and adaptive congestion control based on packet loss and ECN Congestion Experienced (CE) marks. These tests are likely to have complicated interactions with cross traffic and under some conditions can be inversely sensitive to load. For example a test to verify that an AQM algorithm causes ECN CE marks

or packet drops early enough to limit queue occupancy may experience a false pass result in the presence of cross traffic. It is important that engineering tests be performed under a wide range of conditions, including both in situ and bench testing, and over a wide variety of load conditions. Ongoing monitoring is less likely to be useful for engineering tests, although sparse in situ testing might be appropriate.

#### 4. Background

At the time the "Framework for IP Performance Metrics" [RFC2330] was published (1998), sound Bulk Transport Capacity (BTC) measurement was known to be well beyond our capabilities. Even when Framework for Empirical BTC Metrics [RFC3148] was published, we knew that we didn't really understand the problem. Now, by hindsight we understand why assessing BTC is such a hard problem:

- o TCP is a control system with circular dependencies - everything affects performance, including components that are explicitly not part of the test (for example, the host processing power is not in-scope of path performance tests).
- o Congestion control is a dynamic equilibrium process, similar to processes observed in chemistry and other fields. The network and transport protocols find an operating point which balances between opposing forces: the transport protocol pushing harder (raising the data rate and/or window) while the network pushes back (raising packet loss ratio, RTT and/or ECN CE marks). By design TCP congestion control keeps raising the data rate until the network gives some indication that its capacity has been exceeded by dropping packets or adding ECN CE marks. If a TCP sender accurately fills a path to its IP capacity, (e.g. the bottleneck is 100% utilized), then packet losses and ECN CE marks are mostly determined by the TCP sender and how aggressively it seeks additional capacity, and not the network itself, since the network must send exactly the signals that TCP needs to set its rate.
- o TCP's ability to compensate for network impairments (such as loss, delay and delay variation, outside of those caused by TCP itself) is directly proportional to the number of send-ACK round trip exchanges per second (i.e. inversely proportional to the RTT). As a consequence an impaired subpath may pass a short RTT local test even though it fails when the subpath is extended by an effectively perfect network to some larger RTT.
- o TCP has an extreme form of the Observer Effect (colloquially known as the Heisenberg effect). Measurement and cross traffic interact in unknown and ill defined ways. The situation is actually worse than the traditional physics problem where you can at least estimate bounds on the relative momentum of the measurement and measured particles. For network measurement you can not in

general determine even the order of magnitude of the effect. It is possible to construct measurement scenarios where the measurement traffic starves real user traffic, yielding an overly inflated measurement. The inverse is also possible: the user traffic can fill the network, such that the measurement traffic detects only minimal available capacity. You can not in general determine which scenario might be in effect, so you can not gauge the relative magnitude of the uncertainty introduced by interactions with other network traffic.

- o As a consequence of the properties listed above it is difficult, if not impossible, for two independent implementations (HW or SW) of TCP congestion control to produce equivalent performance results [RFC6576] under the same network conditions,

These properties are a consequence of the dynamic equilibrium behavior intrinsic to how all throughput maximizing protocols interact with the Internet. These protocols rely on control systems based on estimated network metrics to regulate the quantity of data to send into the network. The packet sending characteristics in turn alter the network properties estimated by the control system metrics, such that there are circular dependencies between every transmission characteristic and every estimated metric. Since some of these dependencies are nonlinear, the entire system is nonlinear, and any change anywhere causes a difficult to predict response in network metrics. As a consequence Bulk Transport Capacity metrics have not fulfilled the analytic framework envisioned in [RFC2330]

Model Based Metrics overcome these problems by making the measurement system open loop: the packet transfer statistics (akin to the network estimators) do not affect the traffic or traffic patterns (bursts), which are computed on the basis of the Target Transport Performance. A path or subpath meeting the Target Transfer Performance requirements would exhibit packet transfer statistics and estimated metrics that would not cause the control system to slow the traffic below the Target Data Rate.

#### 4.1. TCP properties

TCP and other self clocked protocols (e.g. SCTP) carry the vast majority of all Internet data. Their dominant bulk data transport behavior is to have an approximately fixed quantity of data and acknowledgments (ACKs) circulating in the network. The data receiver reports arriving data by returning ACKs to the data sender, the data sender typically responds by sending approximately the same quantity of data back into the network. The total quantity of data plus the data represented by ACKs circulating in the network is referred to as the window. The mandatory congestion control algorithms incrementally adjust the window by sending slightly more or less data

in response to each ACK. The fundamentally important property of this system is that it is self clocked: The data transmissions are a reflection of the ACKs that were delivered by the network, the ACKs are a reflection of the data arriving from the network.

A number of protocol features cause bursts of data, even in idealized networks that can be modeled as simple queuing systems.

During slowstart the IP rate is doubled on each RTT by sending twice as much data as was delivered to the receiver during the prior RTT. Each returning ACK causes the sender to transmit twice the data the ACK reported arriving at the receiver. For slowstart to be able to fill the pipe, the network must be able to tolerate slowstart bursts up to the full pipe size inflated by the anticipated window reduction on the first loss or ECN CE mark. For example, with classic Reno congestion control, an optimal slowstart has to end with a burst that is twice the bottleneck rate for one RTT in duration. This burst causes a queue which is equal to the pipe size (i.e. the window is twice the pipe size) so when the window is halved in response to the first packet loss, the new window will be the pipe size.

Note that if the bottleneck IP rate is less than half of the capacity of the front path (which is almost always the case), the slowstart bursts will not by themselves cause significant queues anywhere else along the front path; they primarily exercise the queue at the dominant bottleneck.

Several common efficiency algorithms also cause bursts. The self clock is typically applied to groups of packets: the receiver's delayed ACK algorithm generally sends only one ACK per two data segments. Furthermore the modern senders use TCP segmentation offload (TSO) to reduce CPU overhead. The sender's software stack builds super sized TCP segments that the TSO hardware splits into MTU sized segments on the wire. The net effect of TSO, delayed ACK and other efficiency algorithms is to send bursts of segments at full sender interface rate.

Note that these efficiency algorithms are almost always in effect, including during slowstart, such that slowstart typically has a two level burst structure. Section 6.1 describes slowstart in more detail.

Additional sources of bursts include TCP's initial window [RFC6928], application pauses, channel allocation mechanisms and network devices that schedule ACKs. Appendix B describes these last two items. If the application pauses (stops reading or writing data) for some fraction of an RTT, many TCP implementations catch up to their earlier window size by sending a burst of data at the full sender

interface rate. To fill a network with a realistic application, the network has to be able to tolerate sender interface rate bursts large enough to restore the prior window following application pauses.

Although the sender interface rate bursts are typically smaller than the last burst of a slowstart, they are at a higher IP rate so they potentially exercise queues at arbitrary points along the front path from the data sender up to and including the queue at the dominant bottleneck. It is known that these bursts can hurt network performance, especially in conjunction with other queue pressure, however we are not aware of any models for how frequent sender rate bursts the network should be able to tolerate at various burst sizes.

In conclusion, to verify that a path can meet a Target Transport Performance, it is necessary to independently confirm that the path can tolerate bursts at the scales that can be caused by the above mechanisms. Three cases are believed to be sufficient:

- o Two level slowstart bursts sufficient to get connections started properly.
- o Ubiquitous sender interface rate bursts caused by efficiency algorithms. We assume 4 packet bursts to be the most common case, since it matches the effects of delayed ACK during slowstart. These bursts should be assumed not to significantly affect packet transfer statistics.
- o Infrequent sender interface rate bursts that are the maximum of the full target\_window\_size and the initial window size (10 segments in [RFC6928]). The Target\_run\_length may be derated for these large fast bursts.

If a subpath can meet the required packet loss ratio for bursts at all of these scales then it has sufficient buffering at all potential bottlenecks to tolerate any of the bursts that are likely introduced by TCP or other transport protocols.

#### 4.2. Diagnostic Approach

A complete path of a given RTT and MTU, which are equal to or smaller than the Target RTT and equal to or larger than the Target MTU respectively, is expected to be able to attain a specified Bulk Transport Capacity when all of the following conditions are met:

1. The IP capacity is above the Target Data Rate by sufficient margin to cover all TCP/IP overheads. This can be confirmed by the tests described in Section 8.1 or any number of IP capacity tests adapted to implement MBM.
2. The observed packet transfer statistics are better than required by a suitable TCP performance model (e.g. fewer packet losses or

- ECN CE marks). See Section 8.1 or any number of low or fixed rate packet loss tests outside of MBM.
3. There is sufficient buffering at the dominant bottleneck to absorb a slowstart bursts large enough to get the flow out of slowstart at a suitable window size. See Section 8.3.
  4. There is sufficient buffering in the front path to absorb and smooth sender interface rate bursts at all scales that are likely to be generated by the application, any channel arbitration in the ACK path or any other mechanisms. See Section 8.4.
  5. When there is a slowly rising standing queue at the bottleneck the onset of packet loss has to be at an appropriate point (time or queue depth) and progressive [RFC7567]. See Section 8.2.
  6. When there is a standing queue at a bottleneck for a shared media subpath (e.g. half duplex), there must be a suitable bounds on the interaction between ACKs and data, for example due to the channel arbitration mechanism. See Section 8.2.4.

Note that conditions 1 through 4 require capacity tests for validation, and thus may need to be monitored on an ongoing basis. Conditions 5 and 6 require engineering tests, which are best performed in controlled environments such as a bench test. They won't generally fail due to load, but may fail in the field due to configuration errors, etc. and should be spot checked.

A tool that can perform many of the tests is available from [MBMSource].

#### 4.3. New requirements relative to RFC 2330

Model Based Metrics are designed to fulfill some additional requirements that were not recognized at the time RFC 2330 was written [RFC2330]. These missing requirements may have significantly contributed to policy difficulties in the IP measurement space. Some additional requirements are:

- o IP metrics must be actionable by the ISP - they have to be interpreted in terms of behaviors or properties at the IP or lower layers, that an ISP can test, repair and verify.
- o Metrics should be spatially composable, such that measures of concatenated paths should be predictable from subpaths.
- o Metrics must be vantage point invariant over a significant range of measurement point choices, including off path measurement points. The only requirements on MP selection should be that the RTT between the MPs is below some reasonable bound, and that the effects of the "test leads" connecting MPs to the subpath under test can be calibrated out of the measurements. The latter might be accomplished if the test leads are effectively ideal or their properties can be deducted from the measurements between

the MPs. While many of tests require that the test leads have at least as much IP capacity as the subpath under test, some do not, for example Background Packet Transfer Tests described in Section 8.1.3.

- o Metric measurements should be repeatable by multiple parties with no specialized access to MPs or diagnostic infrastructure. It should be possible for different parties to make the same measurement and observe the same results. In particular it is specifically important that both a consumer (or their delegate) and ISP be able to perform the same measurement and get the same result. Note that vantage independence is key to meeting this requirement.

## 5. Common Models and Parameters

### 5.1. Target End-to-end parameters

The target end-to-end parameters are the Target Data Rate, Target RTT and Target MTU as defined in Section 3. These parameters are determined by the needs of the application or the ultimate end user and the complete Internet path over which the application is expected to operate. The target parameters are in units that make sense to upper layers: payload bytes delivered to the application, above TCP. They exclude overheads associated with TCP and IP headers, retransmits and other protocols (e.g. DNS). Note that IP-based network services include TCP headers and retransmissions as part of delivered payload, and this difference is recognized in calculations below (`header_overhead`).

Other end-to-end parameters defined in Section 3 include the effective bottleneck data rate, the sender interface data rate and the TCP and IP header sizes.

The `target_data_rate` must be smaller than all subpath IP capacities by enough headroom to carry the transport protocol overhead, explicitly including retransmissions and an allowance for fluctuations in TCP's actual data rate. Specifying a `target_data_rate` with insufficient headroom is likely to result in brittle measurements having little predictive value.

Note that the target parameters can be specified for a hypothetical path, for example to construct TIDS designed for bench testing in the absence of a real application; or for a live in situ test of production infrastructure.

The number of concurrent connections is explicitly not a parameter to this model. If a subpath requires multiple connections in order to

meet the specified performance, that must be stated explicitly and the procedure described in Section 6.4 applies.

## 5.2. Common Model Calculations

The Target Transport Performance is used to derive the `target_window_size` and the reference `target_run_length`.

The `target_window_size`, is the average window size in packets needed to meet the `target_rate`, for the specified `target_RTT` and `target_MTU`. It is given by:

$$\text{target\_window\_size} = \text{ceiling}(\text{target\_rate} * \text{target\_RTT} / (\text{target\_MTU} - \text{header\_overhead}))$$

`Target_run_length` is an estimate of the minimum required number of unmarked packets that must be delivered between losses or ECN Congestion Experienced (CE) marks, as computed by a mathematical model of TCP congestion control. The derivation here follows [MSM097], and by design is quite conservative.

Reference `target_run_length` is derived as follows: assume the `subpath_IP_capacity` is infinitesimally larger than the `target_data_rate` plus the required `header_overhead`. Then `target_window_size` also predicts the onset of queuing. A larger window will cause a standing queue at the bottleneck.

Assume the transport protocol is using standard Reno style Additive Increase, Multiplicative Decrease (AIMD) congestion control [RFC5681] (but not Appropriate Byte Counting [RFC3465]) and the receiver is using standard delayed ACKs. Reno increases the window by one packet every `pipe_size` worth of ACKs. With delayed ACKs this takes 2 Round Trip Times per increase. To exactly fill the pipe, the spacing of losses must be no closer than when the peak of the AIMD sawtooth reached exactly twice the `target_window_size`. Otherwise, the multiplicative window reduction triggered by the loss would cause the network to be under-filled. Following [MSM097] the number of packets between losses must be the area under the AIMD sawtooth. They must be no more frequent than every 1 in  $((3/2)*\text{target\_window\_size})*(2*\text{target\_window\_size})$  packets, which simplifies to:

$$\text{target\_run\_length} = 3*(\text{target\_window\_size}^2)$$

Note that this calculation is very conservative and is based on a number of assumptions that may not apply. Appendix A discusses these assumptions and provides some alternative models. If a different model is used, a FS-TIDS must document the actual method for



computing `target_run_length` and ratio between alternate `target_run_length` and the reference `target_run_length` calculated above, along with a discussion of the rationale for the underlying assumptions.

These two parameters, `target_window_size` and `target_run_length`, directly imply most of the individual parameters for the tests in Section 8.

### 5.3. Parameter Derating

Since some aspects of the models are very conservative, the MBM framework permits some latitude in derating test parameters. Rather than trying to formalize more complicated models we permit some test parameters to be relaxed as long as they meet some additional procedural constraints:

- o The FS-TIDS must document and justify the actual method used to compute the derated metric parameters.
- o The validation procedures described in Section 10 must be used to demonstrate the feasibility of meeting the Target Transport Performance with infrastructure that just barely passes the derated tests.
- o The validation process for a FS-TIDS itself must be documented in such a way that other researchers can duplicate the validation experiments.

Except as noted, all tests below assume no derating. Tests where there is not currently a well established model for the required parameters explicitly include derating as a way to indicate flexibility in the parameters.

### 5.4. Test Preconditions

Many tests have preconditions which are required to assure their validity. Examples include: the presence or non-presence of cross traffic on specific subpaths; negotiating ECN; and appropriate preamble packet stream to testing to put reactive network elements into the proper states [RFC7312]. If preconditions are not properly satisfied for some reason, the tests should be considered to be inconclusive. In general it is useful to preserve diagnostic information as to why the preconditions were not met, and any test data that was collected even if it is not useful for the intended test. Such diagnostic information and partial test data may be useful for improving the test or test procedures themselves.

It is important to preserve the record that a test was scheduled, because otherwise precondition enforcement mechanisms can introduce

sampling bias. For example, canceling tests due to cross traffic on subscriber access links might introduce sampling bias in tests of the rest of the network by reducing the number of tests during peak network load.

Test preconditions and failure actions must be specified in a FS-TIDS.

## 6. Generating test streams

Many important properties of Model Based Metrics, such as vantage independence, are a consequence of using test streams that have temporal structures that mimic TCP or other transport protocols running over a complete path. As described in Section 4.1, self clocked protocols naturally have burst structures related to the RTT and pipe size of the complete path. These bursts naturally get larger (contain more packets) as either the Target RTT or Target Data Rate get larger, or the Target MTU gets smaller. An implication of these relationships is that test streams generated by running self clocked protocols over short subpaths may not adequately exercise the queuing at any bottleneck to determine if the subpath can support the full Target Transport Performance over the complete path.

Failing to authentically mimic TCP's temporal structure is part of the reason why simple performance tools such as iPerf, netperf, nc, etc have the reputation of yielding false pass results over short test paths, even when some subpath has a flaw.

The definitions in Section 3 are sufficient for most test streams. We describe the slowstart and standing queue test streams in more detail.

In conventional measurement practice stochastic processes are used to eliminate many unintended correlations and sample biases. However MBM tests are designed to explicitly mimic temporal correlations caused by network or protocol elements themselves. Some portions of these systems, such as traffic arrival (test scheduling) are naturally stochastic. Other behaviors, such as back-to-back packet transmissions, are dominated by implementation specific deterministic effects. Although these behaviors always contain non-deterministic elements and might be modeled stochastically, these details typically do not contribute significantly to the overall system behavior. Furthermore, it is known that real protocols are subject to failures caused by network property estimators suffering from bias due to correlation in their own traffic. For example TCP's RTT estimator used to determine the Retransmit Time Out (RTO), can be fooled by periodic cross traffic or start-stop applications. For these reasons many details of the test streams are specified deterministically.

It may prove useful to introduce fine grained noise sources into the models used for generating test streams in an update of Model Based Metrics, but the complexity is not warranted at the time this document was written.

#### 6.1. Mimicking slowstart

TCP slowstart has a two level burst structure as shown in Figure 2. The fine time structure is caused by efficiency algorithms that deliberately batch work (CPU, channel allocation, etc) to better amortize certain network and host overheads. ACKs passing through the return path typically cause the sender to transmit small bursts of data at full sender interface rate. For example TCP Segmentation Offload (TSO) and Delayed Acknowledgment both contribute to this effect. During slowstart these bursts are at the same headway as the returning ACKs, but are typically twice as large (e.g. having twice as much data) as the ACK reported was delivered to the receiver. Due to variations in delayed ACK and algorithms such as Appropriate Byte Counting [RFC3465], different pairs of senders and receivers produce slightly different burst patterns. Without loss of generality, we assume each ACK causes 4 packet sender interface rate bursts at an average headway equal to the ACK headway, and corresponding to sending at an average rate equal to twice the effective bottleneck IP rate. Each slowstart burst consists of a series of 4 packet sender interface rate bursts such that the total number of packets is the current window size (as of the last packet in the burst).

The coarse time structure is due to each RTT being a reflection of the prior RTT. For real transport protocols, each slowstart burst is twice as large (twice the window) as the previous burst but is spread out in time by the network bottleneck, such that each successive RTT exhibits the same effective bottleneck IP rate. The slowstart phase ends on the first lost packet or ECN mark, which is intended to happen after successive slowstart bursts merge in time: the next burst starts before the bottleneck queue is fully drained and the prior burst is complete.

For diagnostic tests described below we preserve the fine time structure but manipulate the coarse structure of the slowstart bursts (burst size and headway) to measure the ability of the dominant bottleneck to absorb and smooth slowstart bursts.

Note that a stream of repeated slowstart bursts has three different average rates, depending on the averaging time interval. At the finest time scale (a few packet times at the sender interface) the peak of the average IP rate is the same as the sender interface rate; at a medium timescale (a few ACK times at the dominant bottleneck) the peak of the average IP rate is twice the implied bottleneck IP

capacity; and at time scales longer than the target\_RTT and when the burst size is equal to the target\_window\_size, the average rate is equal to the target\_data\_rate. This pattern corresponds to repeating the last RTT of TCP slowstart when delayed ACK and sender side byte counting are present but without the limits specified in Appropriate Byte Counting [RFC3465].

time ==> ( - equals one packet)

Fine time structure of the packet stream:

-----

|<>| sender interface rate bursts (typically 3 or 4 packets)  
|<===>| burst headway (from the ACK headway)

\\_\_\_\_\_repeating sender\_\_\_\_\_/   
rate bursts

Coarse (RTT level) time structure of the packet stream:

-----

----- ...

|<=====>| slowstart burst size (from the window)  
|<=====>| slowstart headway (from the RTT)

\\_\_\_\_\_ /   
one slowstart burst

\\_\_\_\_\_ ...   
Repeated slowstart bursts

Multiple levels of Slowstart Bursts

Figure 2

## 6.2. Constant window pseudo CBR

Implement pseudo constant bit rate by running a standard self clocked protocol such as TCP with a fixed window size. If that window size is test\_window, the data rate will be slightly above the target\_rate.

Since the test\_window is constrained to be an integer number of packets, for small RTTs or low data rates there may not be sufficiently precise control over the data rate. Rounding the test\_window up (as defined above) is likely to result in data rates that are higher than the target rate, but reducing the window by one packet may result in data rates that are too small. Also cross traffic potentially raises the RTT, implicitly reducing the rate.

Cross traffic that raises the RTT nearly always makes the test more strenuous (more demanding for the network path).

Note that Constant window pseudo CBR (and Scanned window pseudo CBR in the next section) both rely on a self clock which is at least partially derived from the properties of the subnet under test. This introduces the possibility that the subnet under test exhibits behaviors such as extreme RTT fluctuations that prevent these algorithms from accurately controlling data rates.

A FS-TIDS specifying a constant window CBR test must explicitly indicate under what conditions errors in the data rate cause tests to be inconclusive. Conventional paced measurement traffic may be more appropriate for these environments.

### 6.3. Scanned window pseudo CBR

Scanned window pseudo CBR is similar to the constant window CBR described above, except the window is scanned across a range of sizes designed to include two key events, the onset of queuing and the onset of packet loss or ECN CE marks. The window is scanned by incrementing it by one packet every  $2 \times \text{target\_window\_size}$  delivered packets. This mimics the additive increase phase of standard Reno TCP congestion avoidance when delayed ACKs are in effect. Normally the window increases separated by intervals slightly longer than twice the `target_RTT`.

There are two ways to implement this test: one built by applying a window clamp to standard congestion control in a standard protocol such as TCP and the other built by stiffening a non-standard transport protocol. When standard congestion control is in effect, any losses or ECN CE marks cause the transport to revert to a window smaller than the clamp such that the scanning clamp loses control the window size. The NPAD pathdiag tool is an example of this class of algorithms [Pathdiag].

Alternatively a non-standard congestion control algorithm can respond to losses by transmitting extra data, such that it maintains the specified window size independent of losses or ECN CE marks. Such a stiffened transport explicitly violates mandatory Internet congestion control [RFC5681] and is not suitable for in situ testing. It is only appropriate for engineering testing under laboratory conditions. The Windowed Ping tool implements such a test [WPING]. The tool described in the paper has been updated.[mpingSource]

The test procedures in Section 8.2 describe how to partition the scans into regions and how to interpret the results.

#### 6.4. Concurrent or channelized testing

The procedures described in this document are only directly applicable to single stream measurement, e.g. one TCP connection or measurement stream. In an ideal world, we would disallow all performance claims based multiple concurrent streams, but this is not practical due to at least two issues. First, many very high rate link technologies are channelized and at last partially pin the flow to channel mapping to minimize packet reordering within flows. Second, TCP itself has scaling limits. Although the former problem might be overcome through different design decisions, the later problem is more deeply rooted.

All congestion control algorithms that are philosophically aligned with the standard [RFC5681] (e.g. claim some level of TCP compatibility, friendliness or fairness) have scaling limits, in the sense that as a long fast network (LFN) with a fixed RTT and MTU gets faster, these congestion control algorithms get less accurate and as a consequence have difficulty filling the network [CCscaling]. These properties are a consequence of the original Reno AIMD congestion control design and the requirement in [RFC5681] that all transport protocols have similar responses to congestion.

There are a number of reasons to want to specify performance in terms of multiple concurrent flows, however this approach is not recommended for data rates below several megabits per second, which can be attained with run lengths under 10000 packets on many paths. Since the required run length goes as the square of the data rate, at higher rates the run lengths can be unreasonably large, and multiple flows might be the only feasible approach.

If multiple flows are deemed necessary to meet aggregate performance targets then this must be stated in both the design of the TIDS and in any claims about network performance. The IP diagnostic tests must be performed concurrently with the specified number of connections. For the tests that use bursty test streams, the bursts should be synchronized across streams unless there is a priori knowledge that the applications have some explicit mechanism to stagger their own bursts. In the absences of an explicit mechanism to stagger bursts many network and application artifacts will sometimes implicitly synchronize bursts. A test that does not control burst synchronization may be prone to false pass results for some applications.

## 7. Interpreting the Results

### 7.1. Test outcomes

To perform an exhaustive test of a complete network path, each test of the TIDS is applied to each subpath of the complete path. If any subpath fails any test then a standard transport protocol running over the complete path can also be expected to fail to attain the Target Transport Performance under some conditions.

In addition to passing or failing, a test can be deemed to be inconclusive for a number of reasons. Proper instrumentation and treatment of inconclusive outcomes is critical to the accuracy and robustness of Model Based Metrics. Tests can be inconclusive if the precomputed traffic pattern or data rates were not accurately generated; the measurement results were not statistically significant; and others causes such as failing to meet some required preconditions for the test. See Section 5.4

For example consider a test that implements Constant Window Pseudo CBR (Section 6.2) by adding rate controls and detailed IP packet transfer instrumentation to TCP (e.g. [RFC4898]). TCP includes built in control systems which might interfere with the sending data rate. If such a test meets the required packet transfer statistics (e.g. run length) while failing to attain the specified data rate it must be treated as an inconclusive result, because we can not a priori determine if the reduced data rate was caused by a TCP problem or a network problem, or if the reduced data rate had a material effect on the observed packet transfer statistics.

Note that for capacity tests, if the observed packet transfer statistics meet the statistical criteria for failing (accepting hypothesis H1 in Section 7.2), the test can be considered to have failed because it doesn't really matter that the test didn't attain the required data rate.

The really important new properties of MBM, such as vantage independence, are a direct consequence of opening the control loops in the protocols, such that the test stream does not depend on network conditions or IP packets received. Any mechanism that introduces feedback between the path's measurements and the test stream generation is at risk of introducing nonlinearities that spoil these properties. Any exceptional event that indicates that such feedback has happened should cause the test to be considered inconclusive.

One way to view inconclusive tests is that they reflect situations where a test outcome is ambiguous between limitations of the network

and some unknown limitation of the IP diagnostic test itself, which may have been caused by some uncontrolled feedback from the network.

Note that procedures that attempt to search the target parameter space to find the limits on some parameter such as `target_data_rate` are at risk of breaking the location independent properties of Model Based Metrics, if any part of the boundary between passing and inconclusive or failing results is sensitive to RTT (which is normally the case). For example the maximum data rate for a marginal link (e.g. exhibiting excess errors) is likely to be sensitive to the `test_path_RTT`. The maximum observed data rate over the test path has very little value for predicting the maximum rate over a different path.

One of the goals for evolving TIDS designs will be to keep sharpening distinction between inconclusive, passing and failing tests. The criteria for for passing, failing and inconclusive tests must be explicitly stated for every test in the TIDS or FS-TIDS.

One of the goals of evolving the testing process, procedures, tools and measurement point selection should be to minimize the number of inconclusive tests.

It may be useful to keep raw packet transfer statistics and ancillary metrics [RFC3148] for deeper study of the behavior of the network path and to measure the tools themselves. Raw packet transfer statistics can help to drive tool evolution. Under some conditions it might be possible to re-evaluate the raw data for satisfying alternate Target Transport Performance. However it is important to guard against sampling bias and other implicit feedback which can cause false results and exhibit measurement point vantage sensitivity. Simply applying different delivery criteria based on a different Target Transport Performance is insufficient if the test traffic patterns (bursts, etc.) does not match the alternate Target Transport Performance.

## 7.2. Statistical criteria for estimating `run_length`

When evaluating the observed `run_length`, we need to determine appropriate packet stream sizes and acceptable error levels for efficient measurement. In practice, can we compare the empirically estimated packet loss and ECN Congestion Experienced (CE) marking ratios with the targets as the sample size grows? How large a sample is needed to say that the measurements of packet transfer indicate a particular run length is present?

The generalized measurement can be described as recursive testing: send packets (individually or in patterns) and observe the packet



transfer performance (packet loss ratio or other metric, any marking we define).

As each packet is sent and measured, we have an ongoing estimate of the performance in terms of the ratio of packet loss or ECN CE mark to total packets (i.e. an empirical probability). We continue to send until conditions support a conclusion or a maximum sending limit has been reached.

We have a `target_mark_probability`, 1 mark per `target_run_length`, where a "mark" is defined as a lost packet, a packet with ECN CE mark, or other signal. This constitutes the null Hypothesis:

H0: no more than one mark in `target_run_length` =  
 $3 * (\text{target\_window\_size})^2$  packets

and we can stop sending packets if on-going measurements support accepting H0 with the specified Type I error =  $\alpha$  (= 0.05 for example).

We also have an alternative Hypothesis to evaluate: if performance is significantly lower than the `target_mark_probability`. Based on analysis of typical values and practical limits on measurement duration, we choose four times the H0 probability:

H1: one or more marks in  $(\text{target\_run\_length}/4)$  packets

and we can stop sending packets if measurements support rejecting H0 with the specified Type II error =  $\beta$  (= 0.05 for example), thus preferring the alternate hypothesis H1.

H0 and H1 constitute the Success and Failure outcomes described elsewhere in the memo, and while the ongoing measurements do not support either hypothesis the current status of measurements is inconclusive.

The problem above is formulated to match the Sequential Probability Ratio Test (SPRT) [Wald45] and [Montgomery90]. Note that as originally framed the events under consideration were all manufacturing defects. In networking, ECN CE marks and lost packets are not defects but signals, indicating that the transport protocol should slow down.

The Sequential Probability Ratio Test also starts with a pair of hypothesis specified as above:

H0:  $p_0$  = one defect in `target_run_length`  
H1:  $p_1$  = one defect in `target_run_length/4`

As packets are sent and measurements collected, the tester evaluates the cumulative defect count against two boundaries representing H0 Acceptance or Rejection (and acceptance of H1):

Acceptance line:  $X_a = -h_1 + s \cdot n$

Rejection line:  $X_r = h_2 + s \cdot n$

where  $n$  increases linearly for each packet sent and

$$\begin{aligned} h_1 &= \{ \log((1-\alpha)/\beta) \} / k \\ h_2 &= \{ \log((1-\beta)/\alpha) \} / k \\ k &= \log\{ (p_1(1-p_0)) / (p_0(1-p_1)) \} \\ s &= [ \log\{ (1-p_0)/(1-p_1) \} ] / k \end{aligned}$$

for  $p_0$  and  $p_1$  as defined in the null and alternative Hypotheses statements above, and  $\alpha$  and  $\beta$  as the Type I and Type II errors.

The SPRT specifies simple stopping rules:

- o  $X_a < \text{defect\_count}(n) < X_r$ : continue testing
- o  $\text{defect\_count}(n) \leq X_a$ : Accept H0
- o  $\text{defect\_count}(n) \geq X_r$ : Accept H1

The calculations above are implemented in the R-tool for Statistical Analysis [Rtool] , in the add-on package for Cross-Validation via Sequential Testing (CVST) [CVST].

Using the equations above, we can calculate the minimum number of packets ( $n$ ) needed to accept H0 when  $x$  defects are observed. For example, when  $x = 0$ :

$$\begin{aligned} X_a = 0 &= -h_1 + s \cdot n \\ \text{and } n &= h_1 / s \end{aligned}$$

Note that the derivations in [Wald45] and [Montgomery90] differ. Montgomery's simplified derivation of SPRT may assume a Bernoulli processes, where the packet loss probabilities are independent and identically distributed, making the SPRT more accessible. Wald's seminal paper showed that this assumption is not necessary. It helps to remember that the goal of SPRT is not to estimate the value of the packet loss rate, but only whether or not the packet loss ratio is likely low enough (when we accept the H0 null hypothesis) yielding success; or too high (when we accept the H1 alternate hypothesis) yielding failure.

### 7.3. Reordering Tolerance

All tests must be instrumented for packet level reordering [RFC4737]. However, there is no consensus for how much reordering should be acceptable. Over the last two decades the general trend has been to make protocols and applications more tolerant to reordering (see for example [RFC4015]), in response to the gradual increase in reordering in the network. This increase has been due to the deployment of technologies such as multithreaded routing lookups and Equal Cost MultiPath (ECMP) routing. These techniques increase parallelism in network and are critical to enabling overall Internet growth to exceed Moore's Law.

Note that transport retransmission strategies can trade off reordering tolerance vs how quickly they can repair losses vs overhead from spurious retransmissions. In advance of new retransmission strategies we propose the following strawman: Transport protocols should be able to adapt to reordering as long as the reordering extent is not more than the maximum of one quarter window or 1 mS, whichever is larger. (These values come from experience prototyping Early Retransmit [RFC5827] and related algorithms. They agree with the values being proposed for "RACK: a time-based fast loss detection algorithm" [I-D.ietf-tcpm-rack].) Within this limit on reorder extent, there should be no bound on reordering density.

By implication, recording which is less than these bounds should not be treated as a network impairment. However [RFC4737] still applies: reordering should be instrumented and the maximum reordering that can be properly characterized by the test (because of the bound on history buffers) should be recorded with the measurement results.

Reordering tolerance and diagnostic limitations, such as the size of the history buffer used to diagnose packets that are way out-of-order, must be specified in a FSTIDS.

## 8. IP Diagnostic Tests

The IP diagnostic tests below are organized according to the technique used to generate the test stream as described in Section 6. All of the results are evaluated in accordance with Section 7, possibly with additional test specific criteria.

We also introduce some combined tests which are more efficient when networks are expected to pass, but conflate diagnostic signatures when they fail.

### 8.1. Basic Data Rate and Packet Transfer Tests

We propose several versions of the basic data rate and packet transfer statistics test that differ in how the data rate is controlled. The data can be paced on a timer, or window controlled (and self clocked). The first two tests implicitly confirm that `sub_path` has sufficient raw capacity to carry the `target_data_rate`. They are recommended for relatively infrequent testing, such as an installation or periodic auditing process. The third, background packet transfer statistics, is a low rate test designed for ongoing monitoring for changes in subpath quality.

#### 8.1.1. Delivery Statistics at Paced Full Data Rate

Confirm that the observed run length is at least the `target_run_length` while relying on timer to send data at the `target_rate` using the procedure described in in Section 6.1 with a burst size of 1 (single packets) or 2 (packet pairs).

The test is considered to be inconclusive if the packet transmission can not be accurately controlled for any reason.

RFC 6673 [RFC6673] is appropriate for measuring packet transfer statistics at full data rate.

#### 8.1.2. Delivery Statistics at Full Data Windowed Rate

Confirm that the observed run length is at least the `target_run_length` while sending at an average rate approximately equal to the `target_data_rate`, by controlling (or clamping) the window size of a conventional transport protocol to `test_window`.

Since losses and ECN CE marks cause transport protocols to reduce their data rates, this test is expected to be less precise about controlling its data rate. It should not be considered inconclusive as long as at least some of the round trips reached the full `target_data_rate` without incurring losses or ECN CE marks. To pass this test the network must deliver `target_window_size` packets in `target_RTT` time without any losses or ECN CE marks at least once per two `target_window_size` round trips, in addition to meeting the run length statistical test.

#### 8.1.3. Background Packet Transfer Statistics Tests

The background run length is a low rate version of the target target rate test above, designed for ongoing lightweight monitoring for changes in the observed subpath run length without disrupting users. It should be used in conjunction with one of the above full rate

tests because it does not confirm that the subpath can support raw data rate.

RFC 6673 [RFC6673] is appropriate for measuring background packet transfer statistics.

## 8.2. Standing Queue Tests

These engineering tests confirm that the bottleneck is well behaved across the onset of packet loss, which typically follows after the onset of queuing. Well behaved generally means lossless for transient queues, but once the queue has been sustained for a sufficient period of time (or reaches a sufficient queue depth) there should be a small number of losses or ECN CE marks to signal to the transport protocol that it should reduce its window or data rate. Losses that are too early can prevent the transport from averaging at the `target_data_rate`. Losses that are too late indicate that the queue might not have an appropriate AQM [RFC7567] and as a consequence subject to bufferbloat [wikiBloat]. Queues without AQM have the potential to inflict excess delays on all flows sharing the bottleneck. Excess losses (more than half of the window) at the onset of loss make loss recovery problematic for the transport protocol. Non-linear, erratic or excessive RTT increases suggest poor interactions between the channel acquisition algorithms and the transport self clock. All of the tests in this section use the same basic scanning algorithm, described here, but score the link or subpath on the basis of how well it avoids each of these problems.

Some network technologies rely on virtual queues or other techniques to meter traffic without adding any queuing delay, in which case the data rate will vary with the window size all the way up to the onset of load induced packet loss or ECN CE marks. For these technologies, the discussion of queuing in Section 6.3 does not apply, but it is still necessary to confirm that the onset of losses or ECN CE marks be at an appropriate point and progressive. If the network bottleneck does not introduce significant queuing delay, modify the procedure described in Section 6.3 to start the scan at a window equal to or slightly smaller than the `test_window`.

Use the procedure in Section 6.3 to sweep the window across the onset of queuing and the onset of loss. The tests below all assume that the scan emulates standard additive increase and delayed ACK by incrementing the window by one packet for every  $2 * \text{target\_window\_size}$  packets delivered. A scan can typically be divided into three regions: below the onset of queuing, a standing queue, and at or beyond the onset of loss.

Below the onset of queuing the RTT is typically fairly constant, and the data rate varies in proportion to the window size. Once the data rate reaches the subpath IP rate, the data rate becomes fairly constant, and the RTT increases in proportion to the increase in window size. The precise transition across the start of queuing can be identified by the maximum network power, defined to be the ratio data rate over the RTT. The network power can be computed at each window size, and the window with the maximum is taken as the start of the queuing region.

If there is random background loss (e.g. bit errors, etc), precise determination of the onset of queue induced packet loss may require multiple scans. Above the onset of queuing loss, all transport protocols are expected to experience periodic losses determined by the interaction between the congestion control and AQM algorithms. For standard congestion control algorithms the periodic losses are likely to be relatively widely spaced and the details are typically dominated by the behavior of the transport protocol itself. For the stiffened transport protocols case (with non-standard, aggressive congestion control algorithms) the details of periodic losses will be dominated by how the window increase function responds to loss.

#### 8.2.1. Congestion Avoidance

A subpath passes the congestion avoidance standing queue test if more than `target_run_length` packets are delivered between the onset of queuing (as determined by the window with the maximum network power as described above) and the first loss or ECN CE mark. If this test is implemented using a standard congestion control algorithm with a clamp, it can be performed in situ in the production internet as a capacity test. For an example of such a test see [Pathdiag].

For technologies that do not have conventional queues, use the `test_window` in place of the onset of queuing. i.e. A subpath passes the congestion avoidance standing queue test if more than `target_run_length` packets are delivered between start of the scan at `test_window` and the first loss or ECN CE mark.

#### 8.2.2. Bufferbloat

This test confirms that there is some mechanism to limit buffer occupancy (e.g. that prevents bufferbloat). Note that this is not strictly a requirement for single stream bulk transport capacity, however if there is no mechanism to limit buffer queue occupancy then a single stream with sufficient data to deliver is likely to cause the problems described in [RFC7567], and [wikiBloat]. This may cause only minor symptoms for the dominant flow, but has the potential to make the subpath unusable for other flows and applications.

Pass if the onset of loss occurs before a standing queue has introduced more delay than twice `target_RTT`, or other well defined and specified limit. Note that there is not yet a model for how much standing queue is acceptable. The factor of two chosen here reflects a rule of thumb. In conjunction with the previous test, this test implies that the first loss should occur at a queuing delay which is between one and two times the `target_RTT`.

Specified RTT limits that are larger than twice the `target_RTT` must be fully justified in the FS-TIDS.

#### 8.2.3. Non excessive loss

This test confirms that the onset of loss is not excessive. Pass if losses are equal or less than the increase in the cross traffic plus the test stream window increase since the previous RTT. This could be restated as non-decreasing total throughput of the subpath at the onset of loss. (Note that when there is a transient drop in subpath throughput and there is not already a standing queue, a subpath that passes other queue tests in this document will have sufficient queue space to hold one full RTT worth of data).

Note that token bucket policers will not pass this test, which is as intended. TCP often stumbles badly if more than a small fraction of the packets are dropped in one RTT. Many TCP implementations will require a timeout and slowstart to recover their self clock. Even if they can recover from the massive losses the sudden change in available capacity at the bottleneck wastes serving and front path capacity until TCP can adapt to the new rate [Policing].

#### 8.2.4. Duplex Self Interference

This engineering test confirms a bound on the interactions between the forward data path and the ACK return path when they share a half duplex link.

Some historical half duplex technologies had the property that each direction held the channel until it completely drained its queue. When a self clocked transport protocol, such as TCP, has data and ACKs passing in opposite directions through such a link, the behavior often reverts to stop-and-wait. Each additional packet added to the window raises the observed RTT by two packet times, once as the additional packet passes through the data path, and once for the additional delay incurred by the ACK waiting on the return path.

The duplex self interference test fails if the RTT rises by more than a fixed bound above the expected queuing time computed from the excess window divided by the subpath IP Capacity. This bound must be

smaller than  $\text{target\_RTT}/2$  to avoid reverting to stop and wait behavior. (e.g. Data packets and ACKs both have to be released at least twice per RTT.)

### 8.3. Slowstart tests

These tests mimic slowstart: data is sent at twice the effective bottleneck rate to exercise the queue at the dominant bottleneck.

#### 8.3.1. Full Window slowstart test

This is a capacity test to confirm that slowstart is not likely to exit prematurely. Send slowstart bursts that are  $\text{target\_window\_size}$  total packets.

Accumulate packet transfer statistics as described in Section 7.2 to score the outcome. Pass if it is statistically significant that the observed number of good packets delivered between losses or ECN CE marks is larger than the  $\text{target\_run\_length}$ . Fail if it is statistically significant that the observed interval between losses or ECN CE marks is smaller than the  $\text{target\_run\_length}$ .

It is deemed inconclusive if the elapsed time to send the data burst is not less than half of the time to receive the ACKs. (i.e. It is acceptable to send data too fast, but sending it slower than twice the actual bottleneck rate as indicated by the ACKs is deemed inconclusive). The headway for the slowstart bursts should be the  $\text{target\_RTT}$ .

Note that these are the same parameters as the Sender Full Window burst test, except the burst rate is at slowstart rate, rather than sender interface rate.

#### 8.3.2. Slowstart AQM test

Do a continuous slowstart (send data continuously at twice the implied IP bottleneck capacity), until the first loss, stop, allow the network to drain and repeat, gathering statistics on how many packets were delivered before the loss, the pattern of losses, maximum observed RTT and window size. Justify the results. There is not currently sufficient theory justifying requiring any particular result, however design decisions that affect the outcome of this tests also affect how the network balances between long and short flows (the "mice vs elephants" problem). The queue sojourn time for the first packet delivered after the first loss should be at least one half of the  $\text{target\_RTT}$ .



This is an engineering test: It should be performed on a quiescent network or testbed, since cross traffic has the potential to change the results in ill defined ways.

#### 8.4. Sender Rate Burst tests

These tests determine how well the network can deliver bursts sent at sender's interface rate. Note that this test most heavily exercises the front path, and is likely to include infrastructure may be out of scope for an access ISP, even though the bursts might be caused by ACK compression, thinning or channel arbitration in the access ISP. See Appendix B.

Also, there are a several details about sender interface rate bursts that are not fully defined here. These details, such as the assumed sender interface rate, should be explicitly stated is a FS-TIDS.

Current standards permit TCP to send full window bursts following an application pause. (Congestion Window Validation [RFC2861] and updates to support Rate-Limited Traffic [RFC7661], are not required). Since full window bursts are consistent with standard behavior, it is desirable that the network be able to deliver such bursts, otherwise application pauses will cause unwarranted losses. Note that the AIMD sawtooth requires a peak window that is twice `target_window_size`, so the worst case burst may be  $2 * \text{target\_window\_size}$ .

It is also understood in the application and serving community that interface rate bursts have a cost to the network that has to be balanced against other costs in the servers themselves. For example TCP Segmentation Offload (TSO) reduces server CPU in exchange for larger network bursts, which increase the stress on network buffer memory. Some newer TCP implementations can pace traffic at scale [`TSO_pacing`][`TSO_fq_pacing`]. It remains to be determined if and how quickly these changes will be deployed.

There is not yet theory to unify these costs or to provide a framework for trying to optimize global efficiency. We do not yet have a model for how much server rate bursts should be tolerated by the network. Some bursts must be tolerated by the network, but it is probably unreasonable to expect the network to be able to efficiently deliver all data as a series of bursts.

For this reason, this is the only test for which we encourage derating. A TIDS could include a table of pairs of derating parameters: burst sizes and how much each burst size is permitted to reduce the run length, relative to to the `target_run_length`.

## 8.5. Combined and Implicit Tests

Combined tests efficiently confirm multiple network properties in a single test, possibly as a side effect of normal content delivery. They require less measurement traffic than other testing strategies at the cost of conflating diagnostic signatures when they fail. These are by far the most efficient for monitoring networks that are nominally expected to pass all tests.

### 8.5.1. Sustained Bursts Test

The sustained burst test implements a combined worst case version of all of the capacity tests above. It is simply:

Send `target_window_size` bursts of packets at server interface rate with `target_RTT` burst headway (burst start to next burst start). Verify that the observed packet transfer statistics meets the `target_run_length`.

Key observations:

- o The subpath under test is expected to go idle for some fraction of the time, determined by the difference between the time to drain the queue at the `subpath_IP_capacity`, and the `target_RTT`. If the queue does not drain completely it may be an indication that the subpath has insufficient IP capacity or that there is some other problem with the test (e.g. inconclusive).
- o The burst sensitivity can be derated by sending smaller bursts more frequently. E.g. send `target_window_size*derate` packet bursts every `target_RTT*derate`, where "derate" is less than one.
- o When not derated, this test is the most strenuous capacity test.
- o A subpath that passes this test is likely to be able to sustain higher rates (close to `subpath_IP_capacity`) for paths with RTTs significantly smaller than the `target_RTT`.
- o This test can be implemented with instrumented TCP [RFC4898], using a specialized measurement application at one end [MBMSource] and a minimal service at the other end [RFC0863] [RFC0864].
- o This test is efficient to implement, since it does not require per-packet timers, and can make use of TSO in modern NIC hardware.
- o If a subpath is known to pass the Standing Queue engineering tests (particularly that it has a progressive onset of loss at an appropriate queue depth), then the Sustained Burst Test is sufficient to assure that the subpath under test will not impair Bulk Transport Capacity at the target performance under all conditions. See Section 8.2 for a discussion of the standing queue tests.

Note that this test is clearly independent of the subpath RTT, or other details of the measurement infrastructure, as long as the measurement infrastructure can accurately and reliably deliver the required bursts to the subpath under test.

#### 8.5.2. Passive Measurements

Any non-throughput maximizing application, such as fixed rate streaming media, can be used to implement passive or hybrid (defined in [RFC7799]) versions of Model Based Metrics with some additional instrumentation and possibly a traffic shaper or other controls in the servers. The essential requirement is that the data transmission be constrained such that even with arbitrary application pauses and bursts, the data rate and burst sizes stay within the envelope defined by the individual tests described above.

If the application's serving data rate can be constrained to be less than or equal to the `target_data_rate` and the serving\_RTT (the RTT between the sender and client) is less than the `target_RTT`, this constraint is most easily implemented by clamping the transport window size to `serving_window_clamp`, set to the `test_window`, computed for the actual serving path.

Under the above constraints the `serving_window_clamp` will limit the both the serving data rate and burst sizes to be no larger than the procedures in Section 8.1.2 and Section 8.4 or Section 8.5.1. Since the serving RTT is smaller than the `target_RTT`, the worst case bursts that might be generated under these conditions will be smaller than called for by Section 8.4 and the sender rate burst sizes are implicitly derated by the `serving_window_clamp` divided by the `target_window_size` at the very least. (Depending on the application behavior, the data might be significantly smoother than specified by any of the burst tests.)

In an alternative implementation the data rate and bursts might be explicitly controlled by a programmable traffic shaper or pacing at the sender. This would provide better control over transmissions but is more complicated to implement, although the required technology is available [TSO\_pacing][TSO\_fq\_pacing].

Note that these techniques can be applied to any content delivery that can operate at a constrained data rate to inhibit TCP equilibrium behavior.

Furthermore note that Dynamic Adaptive Streaming over HTTP (DASH) is generally in conflict with passive Model Based Metrics measurement, because it is a rate maximizing protocol. It can still meet the requirement here if the rate can be capped, for example by knowing a

priori the maximum rate needed to deliver a particular piece of content.

## 9. An Example

In this section we illustrate a TIDS designed to confirm that an access ISP can reliably deliver HD video from multiple content providers to all of their customers. With modern codecs, minimal HD video (720p) generally fits in 2.5 Mb/s. Due to their geographical size, network topology and modem characteristics the ISP determines that most content is within a 50 ms RTT of their users (This example RTT is sufficient to cover the propagation delay to continental Europe or either US coast with low delay modems or somewhat smaller geographical regions if the modems require additional delay to implement advanced compression and error recovery).

2.5 Mb/s over a 50 ms path

End-to-End Parameter	value	units
target_rate	2.5	Mb/s
target_RTT	50	ms
target_MTU	1500	bytes
header_overhead	64	bytes
target_window_size	11	packets
target_run_length	363	packets

Table 1

Table 1 shows the default TCP model with no derating, and as such is quite conservative. The simplest TIDS would be to use the sustained burst test, described in Section 8.5.1. Such a test would send 11 packet bursts every 50ms, and confirming that there was no more than 1 packet loss per 33 bursts (363 total packets in 1.650 seconds).

Since this number represents is the entire end-to-end loss budget, independent subpath tests could be implemented by apportioning the packet loss ratio across subpaths. For example 50% of the losses might be allocated to the access or last mile link to the user, 40% to the network interconnections with other ISPs and 1% to each internal hop (assuming no more than 10 internal hops). Then all of the subpaths can be tested independently, and the spatial composition of passing subpaths would be expected to be within the end-to-end loss budget.

### 9.1. Observations about applicability

Guidance on deploying and using MBM belong in a future document. However this example illustrates some the issues that may need to be considered.

Note that another ISP, with different geographical coverage, topology or modem technology may need to assume a different target\_RTT, and as a consequence different target\_window\_size and target\_run\_length, even for the same target\_data rate. One of the implications of this is that infrastructure shared by multiple ISPs, such as inter-exchange points (IXPs) and other interconnects may need to be evaluated on the basis of the most stringent target\_window\_size and target\_run\_length of any participating ISP. One way to do this might be to choose target parameters for evaluating such shared infrastructure on the basis of a hypothetical reference path that does not necessarily match any actual paths.

Testing interconnects has generally been problematic: conventional performance tests run between measurement points adjacent to either side of the interconnect are not generally useful. Unconstrained TCP tests, such as iPerf [iPerf] are usually overly aggressive due to the small RTT (often less than 1 mS). With a short RTT these tools are likely to report inflated data rates because on a short RTT these tools can tolerate very high packet loss ratios and can push other cross traffic off of the network. As a consequence these measurements are useless for predicting actual user performance over longer paths, and may themselves be quite disruptive. Model Based Metrics solves this problem. The interconnect can be evaluated with the same TIDS as other subpaths. Continuing our example, if the interconnect is apportioned 40% of the losses, 11 packet bursts sent every 50mS should have fewer than one loss per 82 bursts (902 packets).

## 10. Validation

Since some aspects of the models are likely to be too conservative, Section 5.2 permits alternate protocol models and Section 5.3 permits test parameter derating. If either of these techniques are used, we require demonstrations that such a TIDS can robustly detect subpaths that will prevent authentic applications using state-of-the-art protocol implementations from meeting the specified Target Transport Performance. This correctness criteria is potentially difficult to prove, because it implicitly requires validating a TIDS against all possible paths and subpaths. The procedures described here are still experimental.

We suggest two approaches, both of which should be applied: first, publish a fully open description of the TIDS, including what assumptions were used and how it was derived, such that the research community can evaluate the design decisions, test them and comment on their applicability; and second, demonstrate that applications do meet the Target Transport Performance when running over a network testbed which has the tightest possible constraints that still allow the tests in the TIDS to pass.

This procedure resembles an epsilon-delta proof in calculus. Construct a test network such that all of the individual tests of the TIDS pass by only small (infinitesimal) margins, and demonstrate that a variety of authentic applications running over real TCP implementations (or other protocols as appropriate) meets the Target Transport Performance over such a network. The workloads should include multiple types of streaming media and transaction oriented short flows (e.g. synthetic web traffic).

For example, for the HD streaming video TIDS described in Section 9, the IP capacity should be exactly the header\_overhead above 2.5 Mb/s, the per packet random background loss ratio should be 1/363, for a run length of 363 packets, the bottleneck queue should be 11 packets and the front path should have just enough buffering to withstand 11 packet interface rate bursts. We want every one of the TIDS tests to fail if we slightly increase the relevant test parameter, so for example sending a 12 packet burst should cause excess (possibly deterministic) packet drops at the dominant queue at the bottleneck. This network has the tightest possible constraints that can be expected to pass the TIDS, yet it should be possible for a real application using a stock TCP implementation in the vendor's default configuration to attain 2.5 Mb/s over an 50 mS path.

The most difficult part of setting up such a testbed is arranging for it to have the tightest possible constraints that still allow it to pass the individual tests. Two approaches are suggested: constraining (configuring) the network devices not to use all available resources (e.g. by limiting available buffer space or data rate); and pre-loading subpaths with cross traffic. Note that it is important that a single tightly constrained environment just barely passes all tests, otherwise there is a chance that TCP can exploit extra latitude in some parameters (such as data rate) to partially compensate for constraints in other parameters (queue space, or vice-versa).

To the extent that a TIDS is used to inform public dialog it should be fully publicly documented, including the details of the tests, what assumptions were used and how it was derived. All of the details of the validation experiment should also be published with

sufficient detail for the experiments to be replicated by other researchers. All components should either be open source or fully described proprietary implementations that are available to the research community.

## 11. Security Considerations

Measurement is often used to inform business and policy decisions, and as a consequence is potentially subject to manipulation. Model Based Metrics are expected to be a huge step forward because equivalent measurements can be performed from multiple vantage points, such that performance claims can be independently validated by multiple parties.

Much of the acrimony in the Net Neutrality debate is due to the historical lack of any effective vantage independent tools to characterize network performance. Traditional methods for measuring Bulk Transport Capacity are sensitive to RTT and as a consequence often yield very different results when run local to an ISP or interconnect and when run over a customer's complete path. Neither the ISP nor customer can repeat the others measurements, leading to high levels of distrust and acrimony. Model Based Metrics are expected to greatly improve this situation.

Note that in situ measurements sometimes requires sending synthetic measurement traffic between arbitrary locations in the network, and as such are potentially attractive platforms for launching DDOS attacks. All active measurement tools and protocols must be designed to minimize the opportunities for these misuses. See the discussion in section 7 of [RFC7594].

Some of the tests described in the note are not intended for frequent network monitoring since they have the potential to cause high network loads and might adversely affect other traffic.

This document only describes a framework for designing Fully Specified Targeted IP Diagnostic Suite. Each FS-TIDS must include its own security section.

## 12. Acknowledgments

Ganga Maguluri suggested the statistical test for measuring loss probability in the target run length. Alex Gilgur and Merry Mou for helping with the statistics.

Meredith Whittaker for improving the clarity of the communications.

Ruediger Geib provided feedback which greatly improved the document.

This work was inspired by Measurement Lab: open tools running on an open platform, using open tools to collect open data. See <http://www.measurementlab.net/>

### 13. IANA Considerations

This document has no actions for IANA.

### 14. Informative References

- [RFC0863] Postel, J., "Discard Protocol", STD 21, RFC 863, May 1983.
- [RFC0864] Postel, J., "Character Generator Protocol", STD 22, RFC 864, May 1983.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2861] Handley, M., Padhye, J., and S. Floyd, "TCP Congestion Window Validation", RFC 2861, June 2000.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC3465] Allman, M., "TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465, February 2003.
- [RFC4015] Ludwig, R. and A. Gurtov, "The Eifel Response Algorithm for TCP", RFC 4015, February 2005.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC4898] Mathis, M., Heffner, J., and R. Raghunarayan, "TCP Extended Statistics MIB", RFC 4898, May 2007.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, February 2008.



- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.
- [RFC5827] Allman, M., Avrachenkov, K., Ayesta, U., Blanton, J., and P. Hurtig, "Early Retransmit for TCP and Stream Control Transmission Protocol (SCTP)", RFC 5827, DOI 10.17487/RFC5827, May 2010, <<http://www.rfc-editor.org/info/rfc5827>>.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.
- [RFC6576] Geib, R., Ed., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, DOI 10.17487/RFC6576, March 2012, <<http://www.rfc-editor.org/info/rfc6576>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, August 2012.
- [RFC6928] Chu, J., Dukkkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, <<http://www.rfc-editor.org/info/rfc6928>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, August 2014.
- [RFC7398] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for Large-Scale Measurement of Broadband Performance", RFC 7398, February 2015.
- [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<http://www.rfc-editor.org/info/rfc7567>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<http://www.rfc-editor.org/info/rfc7594>>.

- [RFC7661] Fairhurst, G., Sathiaselalan, A., and R. Secchi, "Updating TCP to Support Rate-Limited Traffic", RFC 7661, DOI 10.17487/RFC7661, October 2015, <<http://www.rfc-editor.org/info/rfc7661>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<http://www.rfc-editor.org/info/rfc7680>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<http://www.rfc-editor.org/info/rfc7799>>.
- [I-D.ietf-tcpm-rack]  
Cheng, Y., Cardwell, N., and N. Dukkupati, "RACK: a time-based fast loss detection algorithm for TCP", draft-ietf-tcpm-rack-02 (work in progress), March 2017.
- [MSMO97] Mathis, M., Semke, J., Mahdavi, J., and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review volume 27, number3, July 1997.
- [WPING] Mathis, M., "Windowed Ping: An IP Level Performance Diagnostic", INET 94, June 1994.
- [mpingSource]  
Fan, X., Mathis, M., and D. Hamon, "Git Repository for mping: An IP Level Performance Diagnostic", Sept 2013, <<https://github.com/m-lab/mping>>.
- [MBMSource]  
Hamon, D., Stuart, S., and H. Chen, "Git Repository for Model Based Metrics", Sept 2013, <<https://github.com/m-lab/MBM>>.
- [Pathdiag]  
Mathis, M., Heffner, J., O'Neil, P., and P. Siemsen, "Pathdiag: Automated TCP Diagnosis", Passive and Active Measurement , June 2008.
- [iPerf] Wikipedia Contributors, , "iPerf", Wikipedia, The Free Encyclopedia , cited March 2015, <<http://en.wikipedia.org/w/index.php?title=Iperf&oldid=649720021>>.

- [Wald45] Wald, A., "Sequential Tests of Statistical Hypotheses", The Annals of Mathematical Statistics, Vol. 16, No. 2, pp. 117-186, Published by: Institute of Mathematical Statistics, Stable URL: <http://www.jstor.org/stable/2235829>, June 1945.
- [Montgomery90] Montgomery, D., "Introduction to Statistical Quality Control - 2nd ed.", ISBN 0-471-51988-X, 1990.
- [Rtool] R Development Core Team, , "R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>", , 2011.
- [CVST] Krueger, T. and M. Braun, "R package: Fast Cross-Validation via Sequential Testing", version 0.1, 11 2012.
- [AFD] Pan, R., Breslau, L., Prabhakar, B., and S. Shenker, "Approximate fairness through differential dropping", SIGCOMM Comput. Commun. Rev. 33, 2, April 2003.
- [wikiBloat] Wikipedia, , "Bufferbloat", <http://en.wikipedia.org/w/index.php?title=Bufferbloat&oldid=608805474>, March 2015.
- [CCscaling] Fernando, F., Doyle, J., and S. Steven, "Scalable laws for stable network congestion control", Proceedings of Conference on Decision and Control, <http://www.ee.ucla.edu/~paganini>, December 2001.
- [TSO\_pacing] Corbet, J., "TSO sizing and the FQ scheduler", LWN.net <https://lwn.net/Articles/564978/>, Aug 2013.
- [TSO\_fq\_pacing] Dumazet, E. and Y. Chen, "TSO, fair queuing, pacing: three's a charm", Proceedings of IETF 88, TCPM WG <https://www.ietf.org/proceedings/88/slides/slides-88-tcpm-9.pdf>, Nov 2013.
- [Policing] Flach, T., Papageorge, P., Terzis, A., Pedrosa, L., Cheng, Y., Karim, T., Katz-Bassett, E., and R. Govindan, "An Internet-Wide Analysis of Traffic Policing", ACM SIGCOMM , August 2016.

## Appendix A. Model Derivations

The reference `target_run_length` described in Section 5.2 is based on very conservative assumptions: that all excess data in flight (window) above the `target_window_size` contributes to a standing queue that raises the RTT, and that classic Reno congestion control with delayed ACKs are in effect. In this section we provide two alternative calculations using different assumptions.

It may seem out of place to allow such latitude in a measurement method, but this section provides offsetting requirements.

The estimates provided by these models make the most sense if network performance is viewed logarithmically. In the operational Internet, data rates span more than 8 orders of magnitude, RTT spans more than 3 orders of magnitude, and packet loss ratio spans at least 8 orders of magnitude if not more. When viewed logarithmically (as in decibels), these correspond to 80 dB of dynamic range. On an 80 dB scale, a 3 dB error is less than 4% of the scale, even though it represents a factor of 2 in untransformed parameter.

This document gives a lot of latitude for calculating `target_run_length`, however people designing a TIDS should consider the effect of their choices on the ongoing tussle about the relevance of "TCP friendliness" as an appropriate model for Internet capacity allocation. Choosing a `target_run_length` that is substantially smaller than the reference `target_run_length` specified in Section 5.2 strengthens the argument that it may be appropriate to abandon "TCP friendliness" as the Internet fairness model. This gives developers incentive and permission to develop even more aggressive applications and protocols, for example by increasing the number of connections that they open concurrently.

### A.1. Queueless Reno

In Section 5.2 models were derived based on the assumption that the subpath IP rate matches the target rate plus overhead, such that the excess window needed for the AIMD sawtooth causes a fluctuating queue at the bottleneck.

An alternate situation would be a bottleneck where there is no significant queue and losses are caused by some mechanism that does not involve extra delay, for example by the use of a virtual queue as done in Approximate Fair Dropping [AFD]. A flow controlled by such a bottleneck would have a constant RTT and a data rate that fluctuates in a sawtooth due to AIMD congestion control. Assume the losses are being controlled to make the average data rate meet some goal which

is equal or greater than the `target_rate`. The necessary run length to meet the `target_rate` can be computed as follows:

For some value of `Wmin`, the window will sweep from `Wmin` packets to `2*Wmin` packets in `2*Wmin` RTT (due to delayed ACK). Unlike the queuing case where `Wmin = target_window_size`, we want the average of `Wmin` and `2*Wmin` to be the `target_window_size`, so the average data rate is the target rate. Thus we want  $Wmin = (2/3) * target\_window\_size$ .

Between losses each sawtooth delivers  $(1/2)(Wmin + 2*Wmin)(2Wmin)$  packets in `2*Wmin` round trip times.

Substituting these together we get:

$$target\_run\_length = (4/3)(target\_window\_size^2)$$

Note that this is 44% of the `reference_run_length` computed earlier. This makes sense because under the assumptions in Section 5.2 the AMID sawtooth caused a queue at the bottleneck, which raised the effective RTT by 50%.

## Appendix B. The effects of ACK scheduling

For many network technologies simple queuing models don't apply: the network schedules, thins or otherwise alters the timing of ACKs and data, generally to raise the efficiency of the channel allocation algorithms when confronted with relatively widely spaced small ACKs. These efficiency strategies are ubiquitous for half duplex, wireless and broadcast media.

Altering the ACK stream by holding or thinning ACKs typically has two consequences: it raises the implied bottleneck IP capacity, making the fine grained slowstart bursts either faster or larger and it raises the effective RTT by the average time that the ACKs and data are delayed. The first effect can be partially mitigated by re-clocking ACKs once they are beyond the bottleneck on the return path to the sender, however this further raises the effective RTT.

The most extreme example of this sort of behavior would be a half duplex channel that is not released as long as the endpoint currently holding the channel has more traffic (data or ACKs) to send. Such environments cause self clocked protocols under full load to revert to extremely inefficient stop and wait behavior. The channel constrains the protocol to send an entire window of data as a single contiguous burst on the forward path, followed by the entire window of ACKs on the return path.

If a particular return path contains a subpath or device that alters the timing of the ACK stream, then the entire front path from the sender up to the bottleneck must be tested at the burst parameters implied by the ACK scheduling algorithm. The most important parameter is the Implied Bottleneck IP Capacity, which is the average rate at which the ACKs advance `snd.una`. Note that thinning the ACK stream (relying on the cumulative nature of `seg.ack` to permit discarding some ACKs) causes most TCP implementations to send interface rate bursts to offset the longer times between ACKs in order to maintain the average data rate.

Note that due to ubiquitous self clocking in Internet protocols, ill conceived channel allocation mechanisms are likely to increase the queuing stress on the front path because they cause larger full sender rate data bursts.

Holding data or ACKs for channel allocation or other reasons (such as forward error correction) always raises the effective RTT relative to the minimum delay for the path. Therefore it may be necessary to replace `target_RTT` in the calculation in Section 5.2 by an `effective_RTT`, which includes the `target_RTT` plus a term to account for the extra delays introduced by these mechanisms.

#### Appendix C. Version Control

This section to be removed prior to publication.

Formatted: Thu Apr 7 18:12:37 PDT 2016

#### Authors' Addresses

Matt Mathis  
Google, Inc  
1600 Amphitheater Parkway  
Mountain View, California 94043  
USA

Email: [mattmathis@google.com](mailto:mattmathis@google.com)

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

Phone: +1 732 420 1571  
Email: [acmorton@att.com](mailto:acmorton@att.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 9, 2015

A. Morton  
AT&T Labs  
February 5, 2015

Rate Measurement Test Protocol Problem Statement and Requirements  
draft-ietf-ippm-rate-problem-10

Abstract

This memo presents an access rate-measurement problem statement for test protocols to measure IP Performance Metrics. Key rate measurement test protocol aspects include the ability to control packet characteristics on the tested path, such as asymmetric rate and asymmetric packet size.

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 August 9, 2015.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Purpose and Scope . . . . .	3
3. Active Rate Measurement . . . . .	5
4. Measurement Method Categories . . . . .	7
5. Test Protocol Control & Generation Requirements . . . . .	9
6. Security Considerations . . . . .	10
7. Operational Considerations . . . . .	11
8. IANA Considerations . . . . .	11
9. Acknowledgements . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12
10.2. Informative References . . . . .	12
Author's Address . . . . .	13

## 1. Introduction

There are many possible rate measurement scenarios. This memo describes one rate measurement problem and presents a rate-measurement problem statement for test protocols to measure IP Performance Metrics (IPPM).

When selecting a form of access to the Internet, subscribers are interested in the performance characteristics of the various alternatives. Standardized measurements can be a basis for comparison between these alternatives. There is an underlying need to coordinate measurements that support such comparisons, and test control protocols to fulfill this need. The figure below depicts some typical measurement points of access networks.

```

User      /===== Fiber ===== Access Node \
Device -|----- Copper ----- Access Node -|-- Infrastructure -- GW
or Host  \----- Radio ----- Access Node /

```

The access-rate scenario or use case has received wide-spread attention of Internet access subscribers and seemingly all Internet industry players, including regulators. This problem is being approached with many different measurement methods. The eventual protocol solutions to this problem (and the systems that utilize the protocol) may not directly involve users, such as when tests reach



from the Infrastructure to a service-specific device, such as a residential gateway. However, no aspect of the problem precludes users from developing a test protocol controlled via command line interfaces on both ends. Thus, a very wide range of test protocols, active measurement methods and system solutions are the possible outcomes of this problem statement.

## 2. Purpose and Scope

The scope and purpose of this memo is to define the measurement problem statement for test protocols conducting access rate measurement on production networks. Relevant test protocols include [RFC4656] and [RFC5357], but the problem is stated in a general way so that it can be addressed by any existing test protocol, such as [RFC6812].

This memo discusses possibilities for methods of measurement, but does not specify exact methods which would normally be part of the solution, not the problem.

We are interested in access measurement scenarios with the following characteristics:

- o The Access portion of the network is the focus of this problem statement. The user typically subscribes to a service with bi-directional access partly described by rates in bits per second. The rates may be expressed as raw capacity or restricted capacity as described in [RFC6703]. These are the quantities that must be measured according to one or more standard metrics, and for which measurement methods must also be agreed as a part of the solution.
- o Referring to the reference path illustrated below and defined in [I-D.ietf-ippm-lmap-path], possible measurement points include a Subscriber's host, the access service demarcation point, Intra IP access where a globally routable address is present, or the gateway between the measured access network and other networks.

Subsc.	--	Private	--	Private	--	Access	--	Intra IP	--	GRA	--	Transit
device		Net #1		Net #2		Demarc.		Access		GW		GRA GW

GRA = Globally Routable Address, GW = Gateway

- o Rates at some links near the edge of the provider's network can often be several orders of magnitude less than link rates in the aggregation and core portions of the network.
- o Asymmetrical access rates on ingress and egress are prevalent.

- o In many scenarios of interest, extremely large scale of access services requires low complexity devices participating at the user end of the path, and those devices place limits on clock and control timing accuracy.

This problem statement assumes that the most-likely bottleneck device or link is adjacent to the remote (user-end) measurement device, or is within one or two router/switch hops of the remote measurement device.

Other use cases for rate measurement involve situations where the packet switching and transport facilities are leased by one operator from another and the link capacity available cannot be directly determined (e.g., from device interface utilization). These scenarios could include mobile backhaul, Ethernet Service access networks, and/or extensions of layer 2 or layer 3 networks. The results of rate measurements in such cases could be employed to select alternate routing, investigate whether capacity meets some previous agreement, and/or adapt the rate of traffic sources if a capacity bottleneck is found via the rate measurement. In the case of aggregated leased networks, available capacity may also be asymmetric. In these cases, the tester is assumed to have a sender and receiver location under their control. We refer to this scenario below as the aggregated leased network case.

This memo describes protocol support for active measurement methods, consistent with the IPPM working group's traditional charter. Active measurements require synthetic traffic streams dedicated to testing, and do not make measurements on user traffic. See section 2 of [RFC2679], where the concept of a stream is first introduced in IPPM literature as the basis for collecting a sample (defined in section 11 of [RFC2330]).

As noted in [RFC2330] the focus of access traffic management may influence the rate measurement results for some forms of access, as it may differ between user and test traffic if the test traffic has different characteristics, primarily in terms of the packets themselves (see section 13 of [RFC2330] for the considerations on packet type, or Type-P).

There are several aspects of Type-P where user traffic may be examined and selected for special treatment that may affect transmission rates. Various aspects of Type-P are known to influence Equal-Cost Multi-Path (ECMP) routing with possible rate measurement variability across parallel paths. Without being exhaustive, the possibilities include:

- o Packet length

- o IP addresses
- o Transport protocol (e.g. where TCP packets may be routed differently from UDP)
- o Transport Protocol port numbers

This issue requires further discussion when specific solutions/methods of measurement are proposed, but for this problem statement it is sufficient to identify the problem and indicate that the solution may require an extremely close emulation of user traffic, in terms of one or more factors above.

Although the user may have multiple instances of network access available to them, the primary problem scope is to measure one form of access at a time. It is plausible that a solution for the single access problem will be applicable to simultaneous measurement of multiple access instances, but treatment of this scenario is beyond the current scope this document.

A key consideration is whether active measurements will be conducted with user traffic present (In-Service testing), or not present (Out-of-Service testing), such as during pre-service testing or maintenance that interrupts service temporarily. Out-of-Service testing includes activities described as "service commissioning", "service activation", and "planned maintenance". Opportunistic In-Service testing when there is no user traffic present (e.g., outside normal business hours) throughout the test interval is essentially equivalent to Out-of-Service testing. Both In-Service and Out-of-Service testing are within the scope of this problem.

It is a non-goal to solve the measurement protocol specification problem in this memo.

It is a non-goal to standardize methods of measurement in this memo. However, the problem statement mandates support for one category of rate measurement methods in the test protocol and adequate control features for the methods in the control protocol (assuming the control and test protocols are separate).

### 3. Active Rate Measurement

This section lists features of active measurement methods needed to measure access rates in production networks.

Coordination between source and destination devices through control messages and other basic capabilities described in the methods of

IPPM RFCs [RFC2679][RFC2680], and assumed for test protocols such as [RFC5357] and [RFC4656], are taken as given.

Most forms of active testing intrude on user performance to some degree, especially In-Service testing. One key tenet of IPPM methods is to minimize test traffic effects on user traffic in the production network. Section 5 of [RFC2680] lists the problems with high measurement traffic rates ("too much traffic"), and the most relevant for rate measurement is the tendency for measurement traffic to skew the results, followed by the possibility of introducing congestion on the access link. Section 4 of [RFC3148] provides additional considerations. The user of protocols for In-Service testing MUST respect these traffic constraints. Obviously, categories of rate measurement methods that use less active test traffic than others with similar accuracy are preferred for In-Service testing, and the specifications of this memo encourage traffic reduction through asymmetric control capabilities.

Out-of-Service tests where the test path shares no links with In-Service user traffic, have none of the congestion or skew concerns. Both types should address practical matters common to all test efforts, such as conducting measurements within a reasonable time from the tester's point of view, and ensuring that timestamp accuracy is consistent with the precision needed for measurement [RFC2330]. Out-of-Service tests where some part of the test path is shared with In-Service traffic MUST respect the In-Service constraints described above.

The intended metrics to be measured have strong influence over the categories of measurement methods required. For example, using the terminology of [RFC5136], it may be possible to measure a Path Capacity Metric while In-Service if the level of background (user) traffic can be assessed and included in the reported result.

The measurement *\*architecture\** MAY be either of one-way (e.g., [RFC4656]) or two-way (e.g., [RFC5357]), but the scale and complexity aspects of end-user or aggregated access measurement clearly favor two-way (with low-complexity user-end device and round-trip results collection, as found in [RFC5357]). However, the asymmetric rates of many access services mean that the measurement system MUST be able to evaluate performance in each direction of transmission. In the two-way architecture, both end devices MUST include the ability to launch test streams and collect the results of measurements in both (one-way) directions of transmission (this requirement is consistent with previous protocol specifications, and it is not a unique problem for rate measurements).

The following paragraphs describe features for the roles of test packet SENDER, RECEIVER, and results REPORTER.

SENDER:

Generate streams of test packets with various characteristics as desired (see Section 4). The SENDER MAY be located at the user end of the access path or elsewhere in the production network, such as at one end of an aggregated leased network segment.

RECEIVER:

Collect streams of test packets with various characteristics (as described above), and make the measurements necessary to support rate measurement at the receiving end of an access or aggregated leased network segment.

REPORTER:

Use information from test packets and local processes to measure delivered packet rates, and prepare results in the required format (the REPORTER role may be combined with another role, most likely the SENDER).

#### 4. Measurement Method Categories

A protocol that addresses the rate measurement problem MUST serve the test stream generation and measurement functions (SENDER and RECEIVER). The follow-up phase of analyzing the measurement results to produce a report is outside the scope of this problem and memo (REPORTER).

For the purposes of this problem statement, we categorize the many possibilities for rate measurement stream generation as follows;

1. Packet pairs, with fixed intra-pair packet spacing and fixed or random time intervals between pairs in a test stream.
2. Multiple streams of packet pairs, with a range of intra-pair spacing and inter-pair intervals.
3. One or more packet ensembles in a test stream, using a fixed ensemble size in packets and one or more fixed intra-ensemble packet spacings (including zero spacing, meaning that back-to-back burst ensembles and constant rate ensembles fall in this category).

4. One or more packet chirps (a set of packets with specified characteristics), where inter-packet spacing typically decreases between adjacent packets in the same chirp and each pair of packets represents a rate for testing purposes.

The test protocol SHALL support test packet ensemble generation (category 3), as this appears to minimize the demands on measurement accuracy. Other stream generation categories are OPTIONAL.

For all supported categories, the following is a list of additional variables that the protocol(s) MUST be able to specify, control, and generate:

- a. Variable payload lengths among packet streams
- b. Variable length (in packets) among packet streams or ensembles
- c. Variable IP header markings among packet streams
- d. Choice of UDP transport and variable port numbers, OR, choice of TCP transport and variable port numbers for two-way architectures only, OR BOTH. See below for additional requirements on TCP transport generation.
- e. Variable number of packet-pairs, ensembles, or streams used in a test session.

The ability to revise these variables during an established test session is OPTIONAL, as multiple test sessions could serve the same purpose. Another OPTIONAL feature is the ability to generate streams with VLAN tags and other markings.

For measurement systems employing TCP as the transport protocol, the ability to generate specific stream characteristics requires a sender with the ability to establish and prime the connection such that the desired stream characteristics are allowed. See Mathis' work in progress for more background [I-D.ietf-ippm-model-based-metrics].

Beyond simple connection handshake and options establishment, an "open-loop" TCP sender requires the SENDER ability to:

- o generate TCP packets with well-formed headers (all fields valid), including Acknowledgement aspects.
- o produce packet streams at controlled rates and variable inter-packet spacings, including packet ensembles (back-to-back at server rate).

- o continue the configured sending stream characteristics despite all control indications except receive window exhaust.

The corresponding TCP RECEIVER performs normally, having some ability to configure the receive window sufficiently large so as to allow the SENDER to transmit at will (up to a configured target).

It may also be useful (for diagnostic purposes) to provide a control for Bulk Transfer Capacity measurement with fully-specified (and congestion-controlled) TCP senders and receivers, as envisioned in [RFC3148], but this would be a brute-force assessment which does not follow the conservative tenets of IPPM measurement [RFC2330].

Measurements for each UDP test packet transferred between SENDER and RECEIVER MUST be compliant with the singleton measurement methods described in IPPM RFCs [RFC2679][RFC2680]. The time-stamp information or loss/arrival status for each packet MUST be available for communication to the REPORTER function.

## 5. Test Protocol Control & Generation Requirements

In summary, the test protocol must support the measurement features described in the sections above. This requires:

1. Communicating all test variables to the SENDER and RECEIVER
2. Results collection in a one-way architecture
3. Remote device control for both one-way and two-way architectures
4. Asymmetric packet rates in a two-way measurement architecture, or coordinated one-way test capabilities with the same effect (asymmetric rates may be achieved through directional control of packet rate or packet size)

The ability to control and generate asymmetric rates in a two-way architecture is REQUIRED. Two-way architectures are RECOMMENDED to include control and generation capability for both asymmetric and symmetric packet sizes, because packet size often matters in the scope of this problem and test systems SHOULD be equipped to detect directional size dependency through comparative measurements.

Asymmetric packet size control is indicated when the result of a measurement may depend on the size of the packets used in each direction, i.e. when any of the following conditions hold:

- o there is a link in the path with asymmetrical capacity in opposite directions (in combination with one or more of the conditions

below, but their presence or specific details may be unknown to the tester),

- o there is a link in the path which aggregates (or divides) packets into link-level frames, and may have a capacity that depends on packet size, rate, or timing,
- o there is a link in the path where transmission in one direction influences performance in the opposite direction,
- o there is a device in the path where transmission capacity depends on packet header processing capacity (in other words, the capacity is sensitive to packet size),
- o the target application stream is nominally MTU size packets in one direction vs. ACK stream in the other, (noting that there are a vanishing number of symmetrical-rate application streams for which rate measurement is wanted or interesting, but such streams might have some relevance at this time),
- o the distribution of packet losses is critical to rate assessment,

and possibly other circumstances revealed by measurements comparing streams with symmetrical size and asymmetrical size.

Implementations may support control and generation for only symmetric packet sizes when none of the above conditions hold.

The test protocol SHOULD enable measurement of the [RFC5136] Capacity metric, either Out-of-Service, In-Service, or both. Other [RFC5136] metrics are OPTIONAL.

## 6. Security Considerations

The security considerations that apply to any active measurement of live networks are relevant here as well. See [RFC4656] and [RFC5357].

Privacy considerations for measurement systems, particularly when Internet users participate in the tests in some way, are described in [I-D.ietf-lmap-framework].

There may be a serious issue if a proprietary Service Level Agreement involved with the access network segment provider were somehow leaked in the process of rate measurement. To address this, test protocols SHOULD NOT convey this information in a way that could be discovered by unauthorized parties.



## 7. Operational Considerations

All forms of testing originate traffic on the network, through their communications for control and results collection, or from dedicated measurement packet streams, or both. Testing traffic primarily falls in one of two categories, subscriber traffic or network management traffic. There is an on-going need to engineer networks so that various forms of traffic are adequately served, and publication of this memo does not change this need. Service subscribers and authorized users SHOULD obtain their network operator's or service provider's permission before conducting tests. Likewise, a service provider or third party SHOULD obtain the subscriber's permission to conduct tests, since they might temporarily reduce service quality. The protocol SHOULD communicate the permission status once the overall system has obtained it, either explicitly or through other means.

Subscribers, their service providers and network operators, and sometimes third parties, all seek to measure network performance. Capacity testing with active traffic often affects the packet transfer performance of streams traversing shared components of the test path, to some degree. The degradation can be minimized by scheduling such tests infrequently, and restricting the amount of measurement traffic required to assess capacity metrics. As a result, occasional short-duration estimates with minimal traffic are preferred to measurements based on frequent file transfers of many Megabytes with similar accuracy. New measurement methodologies intended for standardization should be evaluated individually for potential operational issues. However, the scheduled frequency of testing is as important as the methods used (and schedules are not typically submitted for standardization).

The new test protocol feature of asymmetrical packet size generation in two-way testing is recommended in this memo. It can appreciably reduce the load and packet processing demands of each test and therefore reduce the likelihood of degradation in one direction of the tested path. Current IETF standardized test protocols (e.g., [RFC5357], also [RFC6812]) do not possess the asymmetric size generation capability with two-way testing.

## 8. IANA Considerations

This memo makes no requests of IANA.

## 9. Acknowledgements

Dave McDysan provided comments and text for the aggregated leased use case. Yaakov Stein suggested many considerations to address, including the In-Service vs. Out-of-Service distinction and its implication on test traffic limits and protocols. Bill Cervený, Marcelo Bagnulo, Kostas Pentikousis (a persistent reviewer), and Joachim Fabini have contributed insightful, clarifying comments that made this a better draft. Barry Constantine also provided suggestions for clarification.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

### 10.2. Informative References

- [I-D.ietf-ippm-lmap-path] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for Large-Scale Measurement of Broadband Performance", draft-ietf-ippm-lmap-path-07 (work in progress), October 2014.

- [I-D.ietf-ippm-model-based-metrics]  
Mathis, M. and A. Morton, "Model Based Bulk Performance Metrics", draft-ietf-ippm-model-based-metrics-03 (work in progress), July 2014.
- [I-D.ietf-lmap-framework]  
Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-10 (work in progress), January 2015.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, February 2008.
- [RFC6812] Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare, S., and E. Yedavalli, "Cisco Service-Level Assurance Protocol", RFC 6812, January 2013.

## Author's Address

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

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

Network Working Group  
Internet-Draft  
Intended status: Best Current Practice  
Expires: August 16, 2014

M. Bagnulo  
UC3M  
B. Claise  
Cisco Systems, Inc.  
P. Eardley  
BT  
A. Morton  
AT&T Labs  
February 12, 2014

Registry for Performance Metrics  
draft-manyfolks-ippm-metric-registry-00

Abstract

This document specifies the common aspects of the IANA registry for performance metrics, both active and passive categories. This document also gives a set of guidelines for Registered Performance Metric requesters and reviewers.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 16, 2014.

Copyright Notice

Copyright (c) 2014 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. Open Issues and Resolutions . . . . .	2
2. Introduction . . . . .	4
3. Terminology . . . . .	5
4. Scope . . . . .	6
5. Design Considerations for the Registry and Registered Metrics	7
5.1. Interoperability . . . . .	7
5.2. Criteria for Registered Performance Metrics . . . . .	8
5.3. Single point of reference for Performance metrics . . . . .	8
5.4. Side benefits . . . . .	9
6. Performance Metric Registry: Prior attempt . . . . .	9
6.1. Why this Attempt Will Succeed? . . . . .	10
7. Common Columns of the Performance Metric Registry . . . . .	10
7.1. Performance Metrics Identifier . . . . .	11
7.2. Performance Metrics Name . . . . .	11
7.3. Performance Metrics Status . . . . .	12
7.4. Performance Metrics Requester . . . . .	12
7.5. Performance Metrics Revision . . . . .	12
7.6. Performance Metrics Revision Date . . . . .	13
7.7. Performance Metrics Description . . . . .	13
7.8. Reference Specification(s) . . . . .	13
8. The Life-Cycle of Registered Metrics . . . . .	13
8.1. The Process for Review by the Performance Metric Experts	13
8.2. Revising Registered Performance Metrics . . . . .	14
8.3. Deprecating Registered Performance Metrics . . . . .	16
9. Performance Metric Registry and other Registries . . . . .	16
10. Security considerations . . . . .	17
11. IANA Considerations . . . . .	17
12. Acknowledgments . . . . .	17
13. References . . . . .	17
13.1. Normative References . . . . .	17
13.2. Informative References . . . . .	18
Authors' Addresses . . . . .	18

## 1. Open Issues and Resolutions

1. I believe that the Performance Metrics Experts and the Performance Metric Directorate will be a different group of people. Reason: every single time a new expert is added, the IESG needs to approve her/him. To be discussed with the Area Directors. \*\*\* (v7) Has this discussion taken place? If these

are different groups, we don't need to define Performance Metrics Directorate.

2. We should expand on the different roles and responsibilities of the Performance Metrics Experts versus the Performance Metric Directorate. At least, the Performance Metric Directorate one should be expanded. --- (v7) If these are different entities, our only concern is the role of the "PM Experts".
3. Not sure if this is interesting for this document to go in the details of the LMAP control protocol versus report protocol (see section 'Interoperability'. (the text currently does this in several sections, S5 comes to mind - Closed)
4. Marcelo, not sure what you mean by 'Single point of reference'. (Closed - see S5.3)
5. Define 'Measurement Parameter'. Even if this is active monitoring specific term, we need it in this draft. Done in v3 Terminology section as "Input Parameter". - Closed in v7 as "Parameter".
6. Performance Metric Description: part of this document of the active/ passive monitoring documents. -- Closed will be Part of Active & Passive docs.
7. Many aspects of the Naming convention are TBD, and need discussion. For example, we have distinguished RTCP-XR metrics as End-Point (neither active nor passive in the traditional sense, so not Act\_ or Pas\_). Also, the Act\_ or Pas\_ component is not consistent with "camel\_case", as Marcelo points out. Even though we may not cast all naming conventions in stone at the start, it will be helpful to look at several examples of passive metric names now.
8. RTCP-XR metrics are currently referred to as "end-point", and have aspects that similar to active (the measured stream characteristics are known a priori and measurement commonly takes place at the end-points of the path) and passive (there is no additional traffic dedicated to measurement, with the exception of the RTCP report packets themselves). We have one example expressing an end-point metric in the active sub-registry memo.
9. Revised Registry Entries: Keep for history (deprecated) or Delete?

10. In section 7 defining the Registry Common Columns, ~all column names begin with "Performance Metric". Al recommends deleting this prefix in each sub-section as redundant.

## 2. Introduction

The IETF specifies and uses Performance Metrics of protocols and applications transported over its protocols. Performance metrics are such an important part of the operations of IETF protocols that [RFC6390] specifies guidelines for their development.

The definition and use of Performance Metrics in the IETF happens in various working groups (WG), most notably:

The "IP Performance Metrics" (IPPM) WG is the WG primarily focusing on Performance Metrics definition at the IETF.

The "Metric Blocks for use with RTCP's Extended Report Framework" (XRBLOCK) WG recently specified many Performance Metrics related to "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611], which establishes a framework to allow new information to be conveyed in RTCP, supplementing the original report blocks defined in "RTP: A Transport Protocol for Real-Time Applications", [RFC3550].

The "Benchmarking Methodology" WG (BMWG) defined many Performance Metrics for use in laboratory benchmarking of inter-networking technologies.

The "IP Flow Information eXport" (IPFIX) WG Information elements related to Performance Metrics are currently proposed.

The "Performance Metrics for Other Layers" (PMOL) concluded WG, defined some Performance Metrics related to Session Initiation Protocol (SIP) voice quality [RFC6035].

It is expected that more Performance Metrics will be defined in the future, not only IP-based metrics, but also metrics which are protocol-specific and application-specific.

However, despite the importance of Performance Metrics, there are two related problems for the industry. First, how to ensure that when one party requests another party to measure (or report or in some way act on) a particular performance metric, then both parties have exactly the same understanding of what performance metric is being referred to. Second, how to discover which Performance Metrics have been specified, so as to avoid developing new performance metric that is very similar. The problems can be addressed by creating a

registry of performance metrics. The usual way in which IETF organizes namespaces is with Internet Assigned Numbers Authority (IANA) registries, and there is currently no Performance Metrics Registry maintained by the IANA.

This document therefore proposes the creation of a Performance Metrics Registry. It also provides best practices on how to define new or updated entries in the Performance Metrics Registry.

### 3. Terminology

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

The terms Performance Metric and Performance Metrics Directorate are defined in [RFC6390], and copied over in this document for the readers convenience.

**Registered Performance Metric:** A Registered Performance Metric (or Registered Metric) is a quantitative measure of performance (see section 6.1 of [RFC2330]) expressed as an entry in the Performance Metric Registry, and comprised of a specifically named metric which has met all the registry review criteria, is under the curation of IETF Performance Metrics Experts, and whose changes are controlled by IANA.

**Registry or Performance Metrics Registry:** The IANA registry containing Registered Performance Metrics.

**Non-IANA Registry:** A set of metrics that are registered locally (and not by IANA).

**Performance Metrics Experts:** The Performance Metrics Experts is a group of experts selected by the IESG to validate the Performance Metrics before updating the Performance Metrics Registry. The Performance Metrics Experts work closely with IANA.

**Performance Metrics Directorate:** The Performance Metrics Directorate is a directorate that provides guidance for Performance Metrics development in the IETF. The Performance Metrics Directorate should be composed of experts in the performance community, potentially selected from the IP Performance Metrics (IPPM), Benchmarking Methodology (BMWG), and Performance Metrics for Other Layers (PMOL) WGs.



**Parameter:** An input factor defined as a variable in the definition of a metric. A numerical or other specified factor forming one of a set that defines a metric or sets the conditions of its operation. Most Input Parameters do not change the fundamental nature of the metric's definition, but others have substantial influence. All Input Parameters must be known to measure using a metric and interpret the results.

**Active Measurement Method:** Methods of Measurement conducted on traffic which serves only the purpose of measurement and is generated for that reason alone, and whose traffic characteristics are known a priori. An Internet user's host can generate active measurement traffic (virtually all typical user-generated traffic is not dedicated to active measurement, but it can produce such traffic with the necessary application operating).

**Passive Measurement Method:** Methods of Measurement conducted on Internet user traffic such that sensitive information is present and may be stored in the measurement system, or observations of traffic from other sources for monitoring and measurement purposes.

**Hybrid Measurement Method:** Methods of Measurement which use a combination of Active Measurement and Passive Measurement methods.

#### 4. Scope

The intended audience of this document includes those who prepare and submit a request for a Registered Performance Metric, and for the Performance Metric Experts who review a request.

This document specifies a Performance Metrics Registry in IANA. This Performance Metric Registry is applicable to Performance Metrics issued from Active Measurement, Passive Measurement, or from end-point calculation. This registry is designed to encompass performance metrics developed throughout the IETF and especially for the following working groups: IPPM, XRBLOCK, IPFIX, BMWG, and possibly others. This document analyzes an prior attempt to set up a Performance Metric Registry, and the reasons why this design was inadequate [RFC6248]. Finally, this document gives a set of guidelines for requesters and expert reviewers of candidate Registered Performance Metrics.

This document serves as the foundation for further work. It specifies the set of columns describing common aspects necessary for all entries in the Performance Metrics Registry.

Two documents describing sub-registries will be developed separately: one for active Registered Metrics and another one for the passive Registered Metrics. Indeed, active and passive performance metrics appear to have different characteristics which must be documented in their respective sub-registries. For example, active performance methods must specify the packet stream characteristics they generate and measure, so it is essential to include the stream specifications in the registry entry. In the case of passive Performance metrics, there is a need to specify the sampling distribution in the registry, while it would be possible to force the definition of the registry field to include both types of distributions in the same registry column, we believe it is cleaner and clearer to have separated sub-registries with different columns that have a narrow definition.

It is possible that future metrics may be a hybrid of active and passive measurement methods, and it may be possible to register hybrid metrics using in one of the two planned sub-registries (active or passive), or it may be efficient to define a third sub-registry with unique columns. The current design with sub-registries allows for growth, and this is a recognized option for extension.

This document makes no attempt to populate the registry with initial entries.

Based on [RFC5226] Section 4.3, this document is processed as Best Current Practice (BCP) [RFC2026].

## 5. Design Considerations for the Registry and Registered Metrics

In this section, we detail several design considerations that are relevant for understanding the motivations and expected use of the metric registry.

### 5.1. Interoperability

As any IETF registry, the primary use for a registry is to manage a namespace for its use within one or more protocols. In this particular case of the metric registry, there are two types of protocols that will use the values defined in the registry for their operation:

- o Control protocol: this type of protocols is used to allow one entity to request another entity to perform a measurement using a specific metric defined by the registry. One particular example is the LMAP framework [I-D.ietf-lmap-framework]. Using the LMAP terminology, the registry is used in the LMAP Control protocol to allow a Controller to request a measurement task to one or more Measurement Agents. In order to enable this use case, the entries

of the metric registry must be well enough defined to allow a Measurement Agent implementation to trigger a specific measurement task upon the reception of a control protocol message. This requirements heavily constrains the type of entries that are acceptable for the Metric registry.

- o Report protocol: This type of protocols is used to allow an entity to report measurement results to another entity. By referencing to a specific metric registry, it is possible to properly characterize the measurement result data being transferred. Using the LMAP terminology, the registry is used in the Report protocol to allow a Measurement Agent to report measurement results to a Collector.

## 5.2. Criteria for Registered Performance Metrics

It is neither possible nor desirable to populate the registry with all combinations of input parameters of all performance metrics. The Registered Performance Metrics should be:

1. interpretable by the user.
2. implementable by the software designer.
3. deployable by network operators, without major impact on the networks.
4. accurate, for interoperability and deployment across vendors

In essence, there needs to be evidence that a candidate registry entry has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Metric serves its intended purpose.

## 5.3. Single point of reference for Performance metrics

A registry for Performance metrics serves as a single point of reference for performance metrics defined in different working groups in the IETF. As we mentioned earlier, there are several WGs that define performance metrics in the IETF and it is hard to keep track of all them. This results in multiple definitions of similar metrics that attempt to measure the same phenomena but in slightly different (and incompatible) ways. Having a registry would allow both the IETF community and external people to have a single list of relevant performance metrics defined by the IETF (and others, where appropriate). The single list is also an essential aspect of communication about metrics, where different entities that request

measurements, execute measurements, and report the results can benefit from a common understanding of the referenced metric.

#### 5.4. Side benefits

There are a couple of side benefits of having such a registry. First, the registry could serve as an inventory of useful and used metrics, that are normally supported by different implementations of measurement agents. Second, the results of the metrics would be comparable even if they are performed by different implementations and in different networks, as the metric is properly defined. BCP 176 [RFC6576] examines whether the results produced by independent implementations are equivalent in the context of evaluating the completeness and clarity of metric specifications. This BCP defines the standards track advancement testing for (active) IPPM metrics, and the same process will likely suffice to determine whether registry entries are sufficiently well specified to result in comparable (or equivalent) results. Registry entries which have undergone such testing SHOULD be noted, with a reference to the test results.

#### 6. Performance Metric Registry: Prior attempt

There was a previous attempt to define a metric registry RFC 4148 [RFC4148]. However, it was obsoleted by RFC 6248 [RFC6248] because it was "found to be insufficiently detailed to uniquely identify IPPM metrics... [there was too much] variability possible when characterizing a metric exactly" which led to the RFC4148 registry having "very few users, if any".

A couple of interesting additional quotes from RFC 6248 might help understand the issues related to that registry.

1. "It is not believed to be feasible or even useful to register every possible combination of Type P, metric parameters, and Stream parameters using the current structure of the IPPM Metrics Registry."
2. "The registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics."
3. "Despite apparent efforts to find current or even future users, no one responded to the call for interest in the RFC 4148 registry during the second half of 2010."

The current approach learns from this by tightly defining each entry in the registry with only a few parameters open, if any. The idea is that entries in the registry represent different measurement methods

which require input parameters to set factors like source and destination addresses (which do not change the fundamental nature of the measurement). The downside of this approach is that it could result in a large number of entries in the registry. We believe that less is more in this context - it is better to have a reduced set of useful metrics rather than a large set of metrics with questionable usefulness. Therefore this document defines that the registry only includes metrics that are well defined and that have proven to be operationally useful. In order to guarantee these two characteristics we require that a set of experts review the allocation request to verify that the metric is well defined and it is operationally useful.

#### 6.1. Why this Attempt Will Succeed?

The registry defined in this document addresses the main issues identified in the previous attempt. As we mention in the previous section, one of the main issues with the previous registry was that the metrics contained in the registry were too generic to be useful. In this registry, the registry requests are evaluated by an expert group that will make sure that the metric is properly defined. This document provides guidelines to assess if a metric is properly defined.

Another key difference between this attempt and the previous one is that in this case there is at least one clear user for the registry: the LMAP framework and protocol. Because the LMAP protocol will use the registry values in its operation, this actually helps to determine if a metric is properly defined. In particular, since we expect that the LMAP control protocol will enable a controller to request a measurement agent to perform a measurement using a given metric by embedding the metric registry value in the protocol, a metric is properly specified if it is defined well-enough so that it is possible (and practical) to implement the metric in the measurement agent. This was clearly not the case for the previous attempt: defining a metric with an undefined P-Type makes its implementation unpractical.

#### 7. Common Columns of the Performance Metric Registry

The metric registry is composed of two sub-registries: the registry for active performance metrics and the registry for passive performance metrics. The rationale for having two sub-registries (as opposed to having a single registry for all metrics) is because the set of registry columns must support unambiguous registry entries, and there are fundamental differences in the methods to collect active and passive metrics and the required input parameters. Forcing them into a single, generalized registry would result in a

less meaningful structure for some entries in the registry. Nevertheless, it is desirable that the two sub-registries share the same structure as much as possible. In particular, both registries will share the following columns: the identifier and the name, the requester, the revision, the revision date and the description. All these fields are described below. The design of these two sub-registries is work-in-progress.

#### 7.1. Performance Metrics Identifier

A numeric identifier for the Registered Performance Metric. This identifier must be unique within the Performance Metric Registry and sub-registries.

The Registered Performance Metric unique identifier is a 16-bit integer (range 0 to 65535). When adding newly Registered Performance Metrics to the Performance Metric Registry, IANA should assign the lowest available identifier to the next active monitoring Registered Performance Metric, and the highest available identifier to the next passive monitoring Registered Performance Metric.

#### 7.2. Performance Metrics Name

As the name of a Registered Performance Metric is the first thing a potential implementor will use when determining whether it is suitable for a given application, it is important to be as precise and descriptive as possible. Names of Registered Performance Metrics:

1. "must be chosen carefully to describe the Registered Performance Metric and the context in which it will be used."
2. "should be unique within the Performance Metric Registry (including sub-registries)."
3. "must use capital letters for the first letter of each component . All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as 'IPv4' and 'IPv6'."
4. "must use '\_' between each component composing the Registered Performance Metric name."
5. "must start with prefix Act\_ for active measurement Registered Performance Metric."

6. "must start with prefix Pass\_ for passive monitoring Registered Performance Metric." AL COMMENTS: how about just 3 letters for consistency: "Pas\_"
7. MARCELO: I am uncertain whether we should give more guidance here for the naming convention. In particular, the second component could be the highest protocol used in the metric (e.g. UDP, TCP, DNS, SIP, ICMP, IPv4, etc). the third component should be a descriptive name (like latency, packet loss or similar). the fourth component could be stream distribution. the fifth component could be the output type (99mean, 95interval). this is of course very active metric oriented, would be good if we could figure out what is the minimum common structure for both passive and active. TBD. AL COMMENTS: Let's see some examples for passive monitoring. It may not make sense to have common name components, except for Act\_ and Pas\_.
8. BENOIT proposes (approximately this, Al's wording) : The remaining rules for naming are left to the Performance Experts to determine as they gather experience, so this is an area of planned update by a future RFC.

An example is "Act\_UDP\_Latency\_Poisson\_99mean" for a active monitoring UDP latency metric using a Poisson stream of packets and producing the 99th percentile mean as output.

>>>> NEED passive naming examples.

#### 7.3. Performance Metrics Status

The status of the specification of this Registered Performance Metric. Allowed values are 'current' and 'deprecated'. All newly defined Information Elements have 'current' status.

#### 7.4. Performance Metrics Requester

The requester for the Registered Performance Metric. The requester may be a document, such as RFC, or person.

#### 7.5. Performance Metrics Revision

The revision number of a Registered Performance Metric, starting at 0 for Registered Performance Metrics at time of definition and incremented by one for each revision.

#### 7.6. Performance Metrics Revision Date

The date of acceptance or the most recent revision for the Registered Performance Metric.

#### 7.7. Performance Metrics Description

A Registered Performance Metric Description is a written representation of a particular registry entry. It supplements the metric name to help registry users select relevant Registered Performance Metrics.

#### 7.8. Reference Specification(s)

Registry entries that follow the common columns must provide the reference specification(s) on which the Registered Performance Metric is based.

### 8. The Life-Cycle of Registered Metrics

Once a Performance Metric or set of Performance Metrics has been identified for a given application, candidate registry entry specifications in accordance with Section X are submitted to IANA to follow the process for review by the Performance Metric Experts, as defined below. This process is also used for other changes to the Performance Metric Registry, such as deprecation or revision, as described later in this section.

It is also desirable that the author(s) of a candidate registry entry seek review in the relevant IETF working group, or offer the opportunity for review on the WG mailing list.

#### 8.1. The Process for Review by the Performance Metric Experts

Requests to change Registered Metrics in the Performance Metric Registry or a linked sub-registry are submitted to IANA, which forwards the request to a designated group of experts (Performance Metric Experts) appointed by the IESG; these are the reviewers called for by the Expert Review RFC5226 policy defined for the Performance Metric Registry. The Performance Metric Experts review the request for such things as compliance with this document, compliance with other applicable Performance Metric-related RFCs, and consistency with the currently defined set of Registered Performance Metrics.

Authors are expected to review compliance with the specifications in this document to check their submissions before sending them to IANA.



The Performance Metric Experts should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the Performance Metric Experts signify their approval to IANA, which changes the Performance Metric Registry. If the request is not acceptable, the Performance Metric Experts can coordinate with the requester to change the request to be compliant. The Performance Metric Experts may also choose in exceptional circumstances to reject clearly frivolous or inappropriate change requests outright.

This process should not in any way be construed as allowing the Performance Metric Experts to overrule IETF consensus. Specifically, any Registered Metrics that were added with IETF consensus require IETF consensus for revision or deprecation.

Decisions by the Performance Metric Experts may be appealed as in Section 7 of RFC5226.

## 8.2. Revising Registered Performance Metrics

Requests to revise the Performance Metric Registry or a linked sub-registry are submitted to IANA, which forwards the request to a designated group of experts (Performance Metric Experts) appointed by the IESG; these are the reviewers called for by the Expert Review [RFC5226] policy defined for the Performance Metric Registry. The Performance Metric Experts review the request for such things as compliance with this document, compliance with other applicable Performance Metric-related RFCs, and consistency with the currently defined set of Registered Performance Metrics.

A request for Revision is ONLY permissible when the changes maintain backward-compatibility with implementations of the prior registry entry describing a Registered Metric (entries with lower revision numbers, but the same Identifier and Name).

The purpose of the Status field in the Performance Metric Registry is to indicate whether the entry for a Registered Metric is 'current' or 'deprecated'.

In addition, no policy is defined for revising IANA Performance Metric entries or addressing errors therein. To be certain, changes and deprecations within the Performance Metric Registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, the provisions of this section address the need for revisions.

Revisions are initiated by sending a candidate Registered Performance Metric definition to IANA, as in Section X, identifying the existing registry entry.

The primary requirement in the definition of a policy for managing changes to existing Registered Performance Metrics is avoidance of interoperability problems; Performance Metric Experts must work to maintain interoperability above all else. Changes to Registered Performance Metrics already in use may only be done in an interoperable way; necessary changes that cannot be done in a way to allow interoperability with unchanged implementations must result in deprecation of the earlier metric.

A change to a Registered Performance Metric is held to be backward-compatible only when:

1. "it involves the correction of an error that is obviously only editorial; or"
2. "it corrects an ambiguity in the Registered Performance Metric's definition, which itself leads to issues severe enough to prevent the Registered Performance Metric's usage as originally defined; or"
3. "it corrects missing information in the metric definition without changing its meaning (e.g., the explicit definition of 'quantity' semantics for numeric fields without a Data Type Semantics value); or"
4. "it harmonizes with an external reference that was itself corrected."
5. "BENOIT: NOTE THAT THERE ARE MORE RULES IN RFC 7013 SECTION 5 BUT THEY WOULD ONLY APPLY TO THE ACTIVE/PASSIVE DRAFTS. TO BE DISCUSSED."

If a change is deemed permissible by the Performance Metric Experts, IANA makes the change in the Performance Metric Registry. The requester of the change is appended to the requester in the registry.

Each Registered Performance Metric in the Registry has a revision number, starting at zero. Each change to a Registered Performance Metric following this process increments the revision number by one.

COMMENT: Al (and Phil) think we should keep old/revised entries as-is, marked as deprecated >>> Since any revision must be interoperable according to the criteria above, there is no need for the Performance Metric Registry to store information about old revisions.

When a revised Registered Performance Metric is accepted into the Performance Metric Registry, the date of acceptance of the most

recent revision is placed into the revision Date column of the registry for that Registered Performance Metric.

Where applicable, additions to registry entries in the form of text Comments or Remarks should include the date, but such additions may not constitute a revision according to this process.

### 8.3. Deprecating Registered Performance Metrics

Changes that are not permissible by the above criteria for Registered Metric's revision may only be handled by deprecation. A Registered Performance Metric MAY be deprecated and replaced when:

1. "the Registered Performance Metric definition has an error or shortcoming that cannot be permissibly changed as in Section Revising Registered Performance Metrics; or"
2. "the deprecation harmonizes with an external reference that was itself deprecated through that reference's accepted deprecation method; or"

A request for deprecation is sent to IANA, which passes it to the Performance Metric Expert for review, as in Section 'The Process for Review by the Performance Metric Experts'. When deprecating an Performance Metric, the Performance Metric description in the Performance Metric Registry must be updated to explain the deprecation, as well as to refer to any new Performance Metrics created to replace the deprecated Performance Metric.

The revision number of a Registered Performance Metric is incremented upon deprecation, and the revision Date updated, as with any revision.

The use of deprecated Registered Metrics should result in a log entry or human-readable warning by the respective application.

Names and Metric ID of deprecated Registered Metrics must not be reused.

## 9. Performance Metric Registry and other Registries

BENOIT: TBD.

THE BASIC IDEA IS THAT PEOPLE COULD DIRECTLY DEFINE PERF. METRICS IN OTHER EXISTING REGISTRIES, FOR SPECIFIC PROTOCOL/ENCODING. EXAMPLE: IPFIX. IDEALLY, ALL PERF. METRICS SHOULD BE DEFINED IN THIS REGISTRY AND REFERS TO FROM OTHER REGISTRIES.

## 10. Security considerations

This draft doesn't introduce any new security considerations for the Internet. However, the definition of Performance Metrics may introduce some security concerns, and should be reviewed with security in mind.

## 11. IANA Considerations

This document specifies the procedure for Performance Metrics Registry setup. IANA is requested to create a new registry for performance metrics called "Registered Performance Metrics".

This Performance Metrics Registry contains two sub registries once for active and another one for passive performance metrics. These sub registries are not defined in this document. However, these two sub registries MUST contain the following columns: the identifier and the name, the requester, the revision, the revision date and the description, as specified in this document.

New assignments for Performance Metric Registry will be administered by IANA through Expert Review [RFC5226], i.e., review by one of a group of experts, the Performance Metric Experts, appointed by the IESG upon recommendation of the Transport Area Directors. The experts will initially be drawn from the Working Group Chairs and document editors of the Performance Metrics Directorate [performance-metrics-directorate].

## 12. Acknowledgments

Thanks to Brian Trammell and Bill Cervený, IPPM chairs, for leading some brainstorming sessions on this topic.

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.

- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.

### 13.2. Informative References

- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC6035] Pendleton, A., Clark, A., Johnston, A., and H. Sinnreich, "Session Initiation Protocol Event Package for Voice Quality Reporting", RFC 6035, November 2010.
- [I-D.ietf-lmap-framework] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-03 (work in progress), January 2014.

### Authors' Addresses

Marcelo Bagnulo  
Universidad Carlos III de Madrid  
Av. Universidad 30  
Leganes, Madrid 28911  
SPAIN

Phone: 34 91 6249500  
Email: marcelo@it.uc3m.es  
URI: <http://www.it.uc3m.es>

Benoit Claise  
Cisco Systems, Inc.  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

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

Philip Eardley  
British Telecom  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: [philip.eardley@bt.com](mailto:philip.eardley@bt.com)

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

Email: [acmorton@att.com](mailto:acmorton@att.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 17, 2014

A. Morton  
AT&T Labs  
M. Bagnulo  
UC3M  
P. Eardley  
BT

February 13, 2014

Active Performance Metric Sub-Registry  
draft-mornuley-ippm-registry-active-00

Abstract

This memo defines the Active Performance Metrics sub-registry of the Performance Metric Registry. This sub-registry will contain Active Performance Metrics, especially those defined in RFCs prepared in the IP Performance Metrics (IPPM) Working Group of the IETF, and possibly applicable to other IETF metrics. Three aspects make IPPM metric registration difficult: (1) Use of the Type-P notion to allow users to specify their own packet types. (2) Use of flexible input variables, called Parameters in IPPM definitions, some of which determine the quantity measured and others of which should not be specified until execution of the measurement. (3) Allowing flexibility in choice of statistics to summarize the results on a stream of measurement packets.

This memo proposes a way to organize registry entries into columns that are well-defined, permitting consistent development of entries over time (a column may be marked NA if it is not applicable for that metric). The design is intended to foster development of registry entries based on existing reference RFCs, whilst each column serves as a check-list item to avoid omissions during the registration process. Every entry in the registry, before IANA action, requires Expert review as defined by concurrent IETF work in progress "Registry for Performance Metrics" (draft-manyfolks-ippm-metric-registry).

The document contains two examples: a registry entry for an active Performance Metric entry based on RFC3393 and RFC5481, and a registry entry for an end-point Performance Metric based on RFC 7003. The examples are for Informational purposes and do not create any entry in the IANA registry.

## 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 August 17, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
1.1. Background and Motivation . . . . .	5
2. Scope . . . . .	7
3. Registry Categories and Columns . . . . .	7
3.1. Common Registry Indexes and Information . . . . .	8
3.1.1. Identifier . . . . .	8
3.1.2. Name . . . . .	8
3.1.3. Status . . . . .	8



3.1.4.	Requester . . . . .	9
3.1.5.	Revision . . . . .	9
3.1.6.	Revision Date . . . . .	9
3.1.7.	Description . . . . .	9
3.1.8.	Reference Specification(s) . . . . .	9
3.2.	Metric Definition . . . . .	9
3.2.1.	Reference Definition . . . . .	9
3.2.2.	Fixed Parameters . . . . .	9
3.3.	Method of Measurement . . . . .	10
3.3.1.	Reference Method . . . . .	10
3.3.2.	Stream Type and Stream Parameters . . . . .	10
3.3.3.	Output Type and Data Format . . . . .	11
3.3.4.	Metric Units . . . . .	11
3.3.5.	Run-time Parameters and Data Format . . . . .	11
3.4.	Comments and Remarks . . . . .	12
4.	Example IPPM Active Registry Entry . . . . .	12
4.1.	Registry Indexes . . . . .	12
4.1.1.	Element ID . . . . .	12
4.1.2.	Metric Name . . . . .	12
4.1.3.	Metric Description . . . . .	12
4.1.4.	Other Info Columns not provided in Example . . . . .	13
4.2.	Metric Definition . . . . .	13
4.2.1.	Reference Definition . . . . .	13
4.2.2.	Fixed Parameters . . . . .	13
4.3.	Method of Measurement . . . . .	13
4.3.1.	Reference Method . . . . .	13
4.3.2.	Stream Type and Stream Parameters . . . . .	13
4.3.3.	Output Type and Data Format . . . . .	14
4.3.4.	Metric Units . . . . .	14
4.3.5.	Run-time Parameters and Data Format . . . . .	14
4.4.	Comments and Remarks . . . . .	15
5.	Example RTCP-XR Registry Entry . . . . .	15
5.1.	Registry Indexes . . . . .	15
5.1.1.	Element ID . . . . .	16
5.1.2.	Metric Name . . . . .	16
5.1.3.	Metric Description . . . . .	16
5.1.4.	Other Info Columns not provided in Example . . . . .	16
5.2.	Metric Definition . . . . .	16
5.2.1.	Reference Definition . . . . .	16
5.2.2.	Fixed Parameters . . . . .	16
5.3.	Method of Measurement . . . . .	17
5.3.1.	Reference Method . . . . .	17
5.3.2.	Stream Type and Stream Parameters . . . . .	17
5.3.3.	Output Type and Data Format . . . . .	18
5.3.4.	Metric Units . . . . .	18
5.3.5.	Run-time Parameters and Data Format . . . . .	18
5.4.	Comments and Remarks . . . . .	20
6.	Example BLANK Registry Entry . . . . .	20

6.1. Registry Indexes . . . . .	20
6.1.1. Element ID . . . . .	20
6.1.2. Metric Name . . . . .	20
6.1.3. Metric Description . . . . .	20
6.1.4. Other Info Columns not provided in Example . . . . .	20
6.2. Metric Definition . . . . .	20
6.2.1. Reference Definition . . . . .	20
6.2.2. Fixed Parameters . . . . .	20
6.3. Method of Measurement . . . . .	21
6.3.1. Reference Method . . . . .	21
6.3.2. Stream Type and Stream Parameters . . . . .	21
6.3.3. Output Type and Data Format . . . . .	21
6.3.4. Metric Units . . . . .	21
6.3.5. Run-time Parameters and Data Format . . . . .	21
6.4. Comments and Remarks . . . . .	22
7. Security Considerations . . . . .	22
8. IANA Considerations . . . . .	22
9. Acknowledgements . . . . .	23
10. References . . . . .	23
10.1. Normative References . . . . .	23
10.2. Informative References . . . . .	24
Authors' Addresses . . . . .	25

## 1. Introduction

### [ISSUES

1. REAL-TIME OR INPUT PARAMETER [CONSISTENT WITH REGISTRY I-D]  
closed - just Parameter
2. CHANGED STREAM PARAMETER TO STREAM INPUT PARAMETER I didn't find  
any instances of this change - closed
3. I PREFER KEEPING THE CATEGORY-COLUMN HIERARCHY - ok we keep it
4. RATHER THAN BLANK COLUMNS, SHOULD WE HAVE 'NOT APPLICABLE' [MAYBE  
EVEN IANA REGISTERED??] sounds good to Al, used NA.
5. THE EXAMPLES ARE INFORMATIONAL NOT STANDARDS TRACK yes of course  
- -Closed.

-----

Note: Efforts to synchronize terminology with  
[I-D.manyfolks-ippm-metric-registry] will likely be incomplete until  
both drafts are stable.

This memo defines the Active Performance Metrics sub-registry of the Performance Metric Registry. This sub-registry will contain Active Performance Metrics, especially those defined in RFCs prepared in the IP Performance Metrics (IPPM) Working Group of the IETF, according to their framework [RFC2330]. Three aspects make IPPM metric registration difficult: (1) Use of the Type-P notion to allow users to specify their own packet types. (2) Use of Flexible input variables, called Parameters in IPPM definitions, some which determine the quantity measured and others which should not be specified until execution of the measurement. (3) Allowing flexibility in choice of statistics to summarize the results on a stream of measurement packets. This memo uses terms and definitions from the IPPM literature, primarily [RFC2330], and the reader is assumed familiar with them or may refer questions there as necessary.

This sub-registry is part of the Performance Metric Registry [I-D.manyfolks-ippm-metric-registry] which specifies that all sub-registries must contain at least the following fields: the identifier, the name, the status, the requester, the revision, the revision date, the description for each entry, and the reference specifications used as the foundation for the Registered Performance Metric (see [I-D.manyfolks-ippm-metric-registry]).

Although there are several standard templates for organizing specifications of performance metrics (see [RFC2679] for an example of the traditional IPPM template, based to large extent on the Benchmarking Methodology Working Group's traditional template in [RFC1242], and see [RFC6390] for a similar template), none of these templates was intended to become the basis for the columns of an IETF-wide registry of metrics. As we examined the aspects of metric specifications which need to be registered, it was clear that none of the existing metric templates fully satisfies the particular needs of a registry.

### 1.1. Background and Motivation

One clear motivation for having such a registry is to allow a controller to request a measurement agent to execute a measurement using a specific metric (see [I-D.ietf-lmap-framework]). Such a request can be performed using any control protocol that refers to the value assigned to the specific metric in the registry. Similarly, the measurement agent can report the results of the measurement and by referring to the metric value it can unequivocally identify the metric that the results correspond to.

There was a previous attempt to define a metric registry RFC 4148 [RFC4148]. However, it was obsoleted by RFC 6248 [RFC6248] because it was "found to be insufficiently detailed to uniquely identify IPPM

metrics... [there was too much] variability possible when characterizing a metric exactly" which led to the RFC4148 registry having "very few users, if any".

Our approach learns from this by tightly defining each entry in the registry with only a few parameters open, if any. The idea is that entries in the registry represent different measurement methods. Each may require run-time parameters to set factors like source and destination addresses, which do not change the fundamental nature of the measurement and can be set just before measurement execution. The downside of this approach is that it could result in a large number of entries in the registry. We believe that less is more in this context - it is better to have a reduced set of useful metrics rather than a large set of metrics with questionable usefulness. Therefore it is required for all registries within the Performance Metric Registry (see [I-D.manyfolks-ippm-metric-registry]) that the registry only includes commonly used metrics that are well defined; hence we require expert review policies for the approval and assignment of entries in this sub-registry.

There are several side benefits of having a registry with well-chosen entries. First, the registry could serve as an inventory of useful and used metrics that are normally supported by different implementations of measurement agents. Second, the results of the metrics would be comparable even if they are performed by different implementations and in different networks, as the metric and method is unambiguously defined.

The registry constitutes a key component of a 'Characterization Plan'. It describes various factors that need to be set by the party controlling the measurements, for example: specific values for the parameters associated with the selected registry entry (for instance, source and destination addresses); and how often the measurement is made. The Characterization Plan determines the individual Measurement Tasks which Measurement Agents will be instructed to do and which they then execute autonomously.

Measurement Instructions might look something like: "Dear measurement agent: Please start test DNS(example.com) and RTT(server.com,150) every day at 2000 GMT. Run the DNS test 5 times and the RTT test 50 times. Do that when the network is idle. Generate both raw results and 99th percentile mean. Send measurement results to collector.com in IPFIX format". The Characterization Plan depends on the requirements of the controlling party. For instance the broadband consumer might want a one-off measurement made immediately to one specific server; a regulator might want the same measurement made once a day until further notice to the 'top 10' servers; whilst an operator might want a varying series of tests (some of which will be

beyond those defined in an IETF registry) as determined from time to time by their operational support system. While the registries defined in this document help to define the Characterization Plan, its full specification falls outside the scope of this document, and other IETF work as currently chartered.

## 2. Scope

[I-D.manyfolks-ippm-metric-registry] defines the overall structure for a Performance Metric Registry and provides guidance for defining a sub registry.

This document defines the Active Performance Metrics Sub-registry; active metrics are those where the packets measured have been specially generated for the purpose.

A row in the registry corresponds to one Registered Performance Metric, with entries in the various columns specifying the metric. Section 3 defines the columns for a Registered Active Performance Metric.

As discussed in [I-D.manyfolks-ippm-metric-registry], each entry (row) must be tightly defined; the definition must leave open only a few parameters that do not change the fundamental nature of the measurement (such as source and destination addresses), and so promotes comparable results across independent implementations. Also, each registered entry must be based on existing reference RFCs (or other standards) for performance metrics, and must be operationally useful and have significant industry interest. This is ensured by expert review for every entry before IANA action.

## 3. Registry Categories and Columns

This section defines the categories and columns of the registry. Below, categories are described at the 3.x heading level, and columns are at the 3.x.y heading level. The Figure below illustrates this organization. An entry (row) therefore gives a complete description of a Registered Metric.

Each column serves as a check-list item and helps to avoid omissions during registration and expert review. In some cases an entry (row) may have some columns without specific entries, marked Not Applicable (NA).

Registry Categories and Columns, shown as

							Category	
							-----	
							Column	Column
							-----	
Common Registry Indexes and Information								
-----								
ID	Name	Status	Request	Rev	Rev.Date	Description	Ref	Spec
-----								
Metric Definition								
-----								
Reference Definition			Fixed Parameters					
-----								
Method of Measurement								
-----								
Reference Method		Stream Type		Output	Output	Run-time		
		and Parameters		Type	Units	Param		
-----								
Comments and Remarks								
-----								

### 3.1. Common Registry Indexes and Information

This category has multiple indexes to each registry entry. It is defined in [I-D.manyfolks-ippm-metric-registry]:

#### 3.1.1. Identifier

Defined in [I-D.manyfolks-ippm-metric-registry]. In order to have the document self contained, we could copy the definition from [I-D.manyfolks-ippm-metric-registry] here, but i guess we should do that once the definition in [I-D.manyfolks-ippm-metric-registry] is stable.

#### 3.1.2. Name

Defined in [I-D.manyfolks-ippm-metric-registry], same comment than above.

#### 3.1.3. Status

Defined in [I-D.manyfolks-ippm-metric-registry], same comment than above.

#### 3.1.4. Requester

Defined in [I-D.manyfolks-ippm-metric-registry], same comment than above.

#### 3.1.5. Revision

Defined in [I-D.manyfolks-ippm-metric-registry], same comment than above.

#### 3.1.6. Revision Date

Defined in [I-D.manyfolks-ippm-metric-registry], same comment than above.

#### 3.1.7. Description

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as the previous.

#### 3.1.8. Reference Specification(s)

Defined in [I-D.manyfolks-ippm-metric-registry], same comment as the previous.

### 3.2. Metric Definition

This category includes columns to prompt all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters, which are left open in the RFC but have a particular value defined by the performance metric.

#### 3.2.1. Reference Definition

This entry provides references to relevant sections of the RFC(s) defining the metric, as well as any supplemental information needed to ensure an unambiguous definition for implementations.

#### 3.2.2. Fixed Parameters

Fixed Parameters are input factors whose value must be specified in the Registry. The measurement system uses these values.

Where referenced metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Fixed Parameters. For example, Fixed Parameters determine most or

all of the IPPM Framework convention "packets of Type-P" as described in [RFC2330], such as transport protocol, payload length, TTL, etc.

A Parameter which is Fixed for one Registry entry may be designated as a Run-time Parameter for another Registry entry.

### 3.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

#### 3.3.1. Reference Method

This entry provides references to relevant sections of the RFC(s) describing the method of measurement, as well as any supplemental information needed to ensure unambiguous interpretation for implementations referring to the RFC text.

#### 3.3.2. Stream Type and Stream Parameters

Principally, two different streams are used in IPPM metrics, Poisson distributed as described in [RFC2330] and Periodic as described in [RFC3432]. Both Poisson and Periodic have their own unique parameters, and the relevant set of values is specified in this column.

Each entry for this column contains the following information:

- o Value: The name of the packet stream scheduling discipline
- o Stream Parameters: The values and formats of input factors for each type of stream. For example, the average packet rate and distribution truncation value for streams with Poisson-distributed inter-packet sending times.
- o Reference: the specification where the stream is defined

The simplest example of stream specification is Singleton scheduling, where a single atomic measurement is conducted. Each atomic measurement could consist of sending a single packet (such as a DNS request) or sending several packets (for example, to request a webpage). Other streams support a series of atomic measurements in a "sample", with a schedule defining the timing between each transmitted packet and subsequent measurement.



### 3.3.3. Output Type and Data Format

For entries which involve a stream and many singleton measurements, a statistic may be specified in this column to summarize the results to a single value. If the complete set of measured singletons is output, this will be specified here.

Some metrics embed one specific statistic in the reference metric definition, while others allow several output types or statistics.

Each entry in the output type column contains the following information:

- o Value: The name of the output type
- o Data Format: provided to simplify the communication with collection systems and implementation of measurement devices.
- o Reference: the specification where the output type is defined

The output type defines the type of result that the metric produces. It can be the raw results or it can be some form of statistic. The specification of the output type must define the format of the output. In some systems, format specifications will simplify both measurement implementation and collection/storage tasks. Note that if two different statistics are required from a single measurement (for example, both "Xth percentile mean" and "Raw"), then a new output type must be defined ("Xth percentile mean AND Raw").

### 3.3.4. Metric Units

The measured results must be expressed using some standard dimension or units of measure. This column provides the units.

When a sample of singletons (see [RFC2330] for definitions of these terms) is collected, this entry will specify the units for each measured value.

### 3.3.5. Run-time Parameters and Data Format

Run-Time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete. However, the values of these parameters is not specified in the Registry, rather these parameters are listed as an aid to the measurement system implementor or user (they must be left as variables, and supplied on execution).

Where metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Run-Time Parameters.

The Data Format of each Run-time Parameter SHALL be specified in this column, to simplify the control and implementation of measurement devices.

Examples of Run-time Parameters include IP addresses, measurement point designations, start times and end times for measurement, and other information essential to the method of measurement.

#### 3.4. Comments and Remarks

Besides providing additional details which do not appear in other categories, this open Category (single column) allows for unforeseen issues to be addressed by simply updating this Informational entry.

#### 4. Example IPPM Active Registry Entry

This section is Informational.

This section gives an example registry entry for the active metric described in [RFC3393], on Packet Delay Variation.

##### 4.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

###### 4.1.1. Element ID

An integer having enough digits to uniquely identify each entry in the Registry.

###### 4.1.2. Metric Name

A metric naming convention is TBD.

One possibility based on IPPM's framework is:

Act\_IP-UDP-One-way-pdv-95th-percentile-Poisson

###### 4.1.3. Metric Description

An assessment of packet delay variation with respect to the minimum delay observed on the stream.

#### 4.1.4. Other Info Columns not provided in Example

#### 4.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

##### 4.2.1. Reference Definition

See sections 2.4 and 3.4 of [RFC3393]. Singleton delay differences measured are referred to by the variable name "ddT".

##### 4.2.2. Fixed Parameters

Since the metric's reference supplies a list of Parameters as part of its descriptive template, a sub-set of the Parameters have been designated as designated as Fixed Parameters for this entry.

- o F, a selection function defining unambiguously the packets from the stream selected for the metric. See section 4.2 of [RFC5481] for the PDV form.
- o L, a packet length in bits. L = 200 bits.
- o Tmax, a maximum waiting time for packets to arrive at Dst, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost). Tmax = 3 seconds.
- o Type-P, as defined in [RFC2330], which includes any field that may affect a packet's treatment as it traverses the network. The packets are IP/UDP, with DSCP = 0 (BE).

#### 4.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

##### 4.3.1. Reference Method

See section 2.6 and 3.6 of [RFC3393] for singleton elements.

##### 4.3.2. Stream Type and Stream Parameters

Poisson distributed as described in [RFC2330], with the following Parameters.

- o  $\lambda$ , a rate in reciprocal seconds (for Poisson Streams).  
 $\lambda = 1$  packet per second
- o Upper limit on Poisson distribution (values above this limit will be clipped and set to the limit value). Upper limit = 30 seconds.

#### 4.3.3. Output Type and Data Format

See section 4.3 of [RFC3393] for details on the percentile statistic.

The percentile = 95.

Data format is a 32-bit unsigned floating point value.

Individual results (singletons) should be represented by the following triple

- o  $T_1$  and  $T_2$ , times as described below in the Run-time parameters section.
- o  $ddT$  as defined in section 2.4 of [RFC3393]

if needed. The result format for  $ddT$  is \*similar to\* the short format in [RFC5905] (32 bits) and is as follows: the first 16 bits represent the \*signed\* integer number of seconds; the next 16 bits represent the fractional part of a second.

#### 4.3.4. Metric Units

See section 3.3 of [RFC3393] for singleton elements.

[RFC2330] recommends that when a time is given, it will be expressed in UTC.

The timestamp format (for  $T$ ,  $T_f$ , etc.) is the same as in [RFC5905] (64 bits) and is as follows: the first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the next 32 bits represent the fractional part of a second that has elapsed since then.

#### 4.3.5. Run-time Parameters and Data Format

Since the metric's reference supplies a list of Parameters as part of its descriptive template, a sub-set of the Parameters have been designated as Run-Time Parameters for this entry. In related registry entries, some of the parameters below may be designated as Fixed Parameters instead.

- o Src, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o Dst, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o T, a time (start of test interval, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o Tf, a time (end of test interval, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o T1, the wire time of the first packet in a pair, measured at MP(Src) as it leaves for Dst (64-bit NTP Timestamp Format, see section 6 of [RFC5905]).
- o T2, the wire time of the second packet in a pair, measured at MP(Src) as it leaves for Dst (64-bit NTP Timestamp Format, see section 6 of [RFC5905]).
- o I(i), I(i+1),  $i \geq 0$ , pairs of times which mark the beginning and ending of the intervals in which the packet stream from which the measurement is taken occurs. Here,  $I(0) = T_0$  and assuming that n is the largest index,  $I(n) = T_f$  (pairs of 64-bit NTP Timestamp Format, see section 6 of [RFC5905]).

#### 4.4. Comments and Remarks

Lost packets represent a challenge for delay variation metrics. See section 4.1 of [RFC3393] and the delay variation applicability statement [RFC5481] for extensive analysis and comparison of PDV and an alternate metric, IPDV.

#### 5. Example RTCP-XR Registry Entry

This section is Informational.

This section gives an example registry entry for the end-point metric described in RFC 7003 [RFC7003], for RTCP-XR Burst/Gap Discard Metric reporting.

##### 5.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

#### 5.1.1. Element ID

An integer having enough digits to uniquely identify each entry in the Registry.

#### 5.1.2. Metric Name

A metric naming convention is TBD.

#### 5.1.3. Metric Description

TBD.

#### 5.1.4. Other Info Columns not provided in Example

### 5.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters. Section 3.2 of [RFC7003] provides the reference information for this category.

#### 5.2.1. Reference Definition

Packets Discarded in Bursts:

The total number of packets discarded during discard bursts. The measured value is unsigned value. If the measured value exceeds 0xFFFFFD, the value 0xFFFFFE MUST be reported to indicate an over-range measurement. If the measurement is unavailable, the value 0xFFFFF MUST be reported.

#### 5.2.2. Fixed Parameters

Fixed Parameters are input factors that must be determined and embedded in the measurement system for use when needed. The values of these parameters is specified in the Registry.

Threshold: 8 bits, set to value = 3 packets.

The Threshold is equivalent to Gmin in [RFC3611], i.e., the number of successive packets that must not be discarded prior to and following a discard packet in order for this discarded packet to be regarded as part of a gap. Note that the Threshold is set in accordance with the Gmin calculation defined in Section 4.7.2 of [RFC3611].

Interval Metric flag: 2 bits, set to value 11=Cumulative Duration

This field is used to indicate whether the burst/gap discard metrics are Sampled, Interval, or Cumulative metrics [RFC6792]:

I=10: Interval Duration - the reported value applies to the most recent measurement interval duration between successive metrics reports.

I=11: Cumulative Duration - the reported value applies to the accumulation period characteristic of cumulative measurements.

Senders MUST NOT use the values I=00 or I=01.

### 5.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations. For the Burst/Gap Discard Metric, it appears that the only guidance on methods of measurement is in Section 3.0 of [RFC7003] and its supporting references. Relevant information is repeated below, although there appears to be no section titled "Method of Measurement" in [RFC7003].

#### 5.3.1. Reference Method

Metrics in this block report on burst/gap discard in the stream arriving at the RTP system. Measurements of these metrics are made at the receiving end of the RTP stream. Instances of this metrics block use the synchronization source (SSRC) to refer to the separate auxiliary Measurement Information Block [RFC6776], which describes measurement periods in use (see [RFC6776], Section 4.2).

This metrics block relies on the measurement period in the Measurement Information Block indicating the span of the report. Senders MUST send this block in the same compound RTCP packet as the Measurement Information Block. Receivers MUST verify that the measurement period is received in the same compound RTCP packet as this metrics block. If not, this metrics block MUST be discarded.

#### 5.3.2. Stream Type and Stream Parameters

Since RTCP-XR Measurements are conducted on live RTP traffic, the complete description of the stream is contained in SDP messages that proceed the establishment of a compatible stream between two or more communicating hosts. See Run-time Parameters, below.

### 5.3.3. Output Type and Data Format

The output type defines the type of result that the metric produces.

- o Value: Packets Discarded in Bursts
- o Data Format: 24 bits
- o Reference: Section 3.2 of [RFC7003]

### 5.3.4. Metric Units

The measured results are apparently expressed in packets, although there is no section of [RFC7003] titled "Metric Units".

### 5.3.5. Run-time Parameters and Data Format

Run-Time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete. However, the values of these parameters is not specified in the Registry, rather these parameters are listed as an aid to the measurement system implementor or user (they must be left as variables, and supplied on execution).

The Data Format of each Run-time Parameter SHALL be specified in this column, to simplify the control and implementation of measurement devices.

SSRC of Source: 32 bits As defined in Section 4.1 of [RFC3611].

SDP Parameters: As defined in [RFC4566]

Session description v= (protocol version number, currently only 0)

o= (originator and session identifier : username, id, version number, network address)

s= (session name : mandatory with at least one UTF-8-encoded character)

i=\* (session title or short information) u=\* (URI of description)

e=\* (zero or more email address with optional name of contacts)

p=\* (zero or more phone number with optional name of contacts)

c=\* (connection information--not required if included in all media)



b=\* (zero or more bandwidth information lines) One or more Time descriptions ("t=" and "r=" lines; see below)

z=\* (time zone adjustments)

k=\* (encryption key)

a=\* (zero or more session attribute lines)

Zero or more Media descriptions (each one starting by an "m=" line; see below)

m= (media name and transport address)

i=\* (media title or information field)

c=\* (connection information -- optional if included at session level)

b=\* (zero or more bandwidth information lines)

k=\* (encryption key)

a=\* (zero or more media attribute lines -- overriding the Session attribute lines)

An example Run-time SDP description follows:

v=0

o=jdoe 2890844526 2890842807 IN IP4 192.0.2.5

s=SDP Seminar i=A Seminar on the session description protocol

u=http://www.example.com/seminars/sdp.pdf e=j.doe@example.com (Jane Doe)

c=IN IP4 233.252.0.12/127

t=2873397496 2873404696

a=recvonly

m=audio 49170 RTP/AVP 0

m=video 51372 RTP/AVP 99

a=rtpmap:99 h263-1998/90000

#### 5.4. Comments and Remarks

TBD.

#### 6. Example BLANK Registry Entry

This section is Informational. (?)

This section gives an example registry entry for the <type of metric and specification reference> .

##### 6.1. Registry Indexes

This category includes multiple indexes to the registry entries, the element ID and metric name.

###### 6.1.1. Element ID

An integer having enough digits to uniquely identify each entry in the Registry.

###### 6.1.2. Metric Name

A metric naming convention is TBD.

###### 6.1.3. Metric Description

A metric Description is TBD.

###### 6.1.4. Other Info Columns not provided in Example

##### 6.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

<possible section reference>.

###### 6.2.1. Reference Definition

###### 6.2.2. Fixed Parameters

Fixed Parameters are input factors that must be determined and embedded in the measurement system for use when needed. The values of these parameters is specified in the Registry.

<list fixed parameters>

### 6.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 6.3.1. Reference Method

For <metric>.

<section reference>

#### 6.3.2. Stream Type and Stream Parameters

<list of stream parameters>.

<references>

#### 6.3.3. Output Type and Data Format

The output type defines the type of result that the metric produces.

- o Value:

- o Data Format: (There may be some precedent to follow here, but otherwise use 64-bit NTP Timestamp Format, see section 6 of [RFC5905]).

- o Reference: <section reference>

#### 6.3.4. Metric Units

The measured results are expressed in <units>.

<section reference>.

#### 6.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

<list of run-time parameters>

<reference(s)>.

#### 6.4. Comments and Remarks

Additional (Informational) details for this entry

#### 7. Security Considerations

This registry has no known implications on Internet Security.

#### 8. IANA Considerations

IANA is requested to create The Active Performance Metric Sub-registry within the Performance Metric Registry defined in [I-D.manyfolks-ippm-metric-registry]. The Sub-registry will contain the following categories and (bullet) columns, (as defined in section 3 above):

Common Registry Indexes and Info

- o Identifier
- o Name
- o Status
- o Requester
- o Revision
- o Revision Date
- o Description
- o Reference Specification(s)

Metric Definition

- o Reference Definition
- o Fixed Parameters

Method of Measurement

- o Reference Method
- o Stream Type and Parameters
- o Output type and Data format

- o Metric Units
- o Run-time Parameters

Comments and Remarks

## 9. Acknowledgements

The authors thank Brian Trammell for suggesting the term "Run-time Parameters", which led to the distinction between run-time and fixed parameters implemented in this memo, and the IPFIX metric with Flow Key as an example.

## 10. References

### 10.1. Normative References

- [I-D.manyfolks-ippm-metric-registry]  
Bagnulo, M., Claise, B., Eardley, P., and A. Morton,  
"Registry for Performance Metrics", Internet Draft (work  
in progress) draft-manyfolks-ippm-metric-registry, 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis,  
"Framework for IP Performance Metrics", RFC 2330, May  
1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way  
Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way  
Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip  
Delay Metric for IPPM", RFC 2681, September 1999.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation  
Metric for IP Performance Metrics (IPPM)", RFC 3393,  
November 2002.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network  
performance measurement with periodic streams", RFC 3432,  
November 2002.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

## 10.2. Informative References

- [Brow00] Brownlee, N., "Packet Matching for NeTraMet Distributions", March 2000.
- [I-D.ietf-lmap-framework] Eardley, P., Morton, A., Bagnulo, M., Burbidge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-03 (work in progress), January 2014.
- [RFC1242] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC 5472, March 2009.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.

[RFC7003] Clark, A., Huang, R., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Discard Metric Reporting", RFC 7003, September 2013.

Authors' Addresses

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

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

Marcelo Bagnulo  
Universidad Carlos III de Madrid  
Av. Universidad 30  
Leganes, Madrid 28911  
SPAIN

Phone: 34 91 6249500  
Email: [marcelo@it.uc3m.es](mailto:marcelo@it.uc3m.es)  
URI: <http://www.it.uc3m.es>

Philip Eardley  
British Telecom  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: [philip.eardley@bt.com](mailto:philip.eardley@bt.com)

IP Performance Measurement (ippm)  
Internet-Draft  
Intended status: Informational  
Expires: August 18, 2014

B. Trammell  
ETH Zurich  
L. Zheng  
Huawei  
S. Silva  
LACNIC  
M. Bagnulo  
UC3M  
February 14, 2014

Hybrid Measurement using IPPM Metrics  
draft-trammell-ippm-hybrid-ps-01

Abstract

Hybrid measurement is the combination of metrics derived from passive and active measurement to produce a measurement result. This document discusses use cases for hybrid measurement using metrics defined within the IPPM framework, and discusses requirements for the definition of passive methodologies for selected IPPM metrics.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 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.

## 1. Introduction

Hybrid measurement is the combination of metrics derived from passive and active measurement to produce a measurement result. This combination can be either spatial or temporal. For example, one way delay to a given endpoint could be derived from passive measurements from a sample of remote endpoints with which traffic is frequently exchanged, and supplemented with active measurements from endpoints with less frequent traffic, to build a "delay map" to a certain point in the network. On the temporal side, loss or delay metrics could be passively measured and stored over time to provide a baseline against which actively-measured loss or delay metrics could be compared during troubleshooting, in order to determine whether a specific path or path segment is contributing to an observed performance problem.

The IPPM working group has produced a framework [RFC2330] for and rich set of well-defined metrics (e.g. [RFC2679], [RFC2680]) for IP performance measurement using active methods, and protocols for measuring them. These metrics could form the basis of a platform for hybrid measurement, provided that passively-derived metrics were defined to be compatible with the corresponding actively-derived metrics; or alternately, provided that methodologies for passive measurement can be defined for each of the existing active metrics to be used, such that those methodologies produce values for the metrics equivalent to the active methodology for the same metric parameters, given some assumptions about the packet stream to be observed to perform the passive measurement, and given tolerances for uncertainty in the results.

## 2. Problem Statement

Complicating the definition of hybrid measurements is that passive measurement must make do with the traffic that is observable, while active measurement has some control over the traffic observed. Measurements for some set of parameters are not possible if no suitable traffic is observed, and the timing of the measurement cannot be controlled. Placement of the observation points for passive measurement along a path additionally introduces uncertainty in the results. For example, passive one-way delay measurement could be performed using two measurement points, one close to each endpoint, with synchronized clocks, comparing the observation times of packets via their hashes. This will not produce a value which is

directly comparable to a Type-P-One-way-Delay measured as specified in section 3.6 of [RFC2679], because it will not account for the one-way-delay from the source to the source-side observation point, or from the destination-side observation point to the destination. Any specification of hybrid measurement using IPPM metrics must handle these complications.

The proposed specification entails:

- o Definition of scenarios and requirements for hybrid measurement.
- o Selection of existing IPPM metrics to be used for the active side of hybrid measurements to meet these requirements.
- o Definition of equivalent passive measurement methodologies for each selected metric, specifically addressing the assumptions about the observed packet stream which must hold for the metric to be valid, and with a specific allowance for the measurement and/or estimation of uncertainty due to uncontrollable conditions or observation point placement.
- o Definition of metrics based on these passive methodologies, or modification of the definition of existing metrics to add passive methodologies.
- o Definition of methods for comparison between active and passive metrics allowing for estimated uncertainty.
- o Definition of methods for spatial and temporal composition of active and passive metrics together allowing for estimated uncertainty.

### 3. Selected IPPM Metrics

[EDITOR'S NOTE: this section will contain information on the metrics selected for passive measurement, and initial discussion of passive measurement methodologies for them. Metric definition will presumably be left for a future document.]

#### 3.1. Packet Loss

In order to perform packet loss measurements on a live traffic flow, different approaches exist. An approach is to count the number of packets sent on one end, and the number of packets received on the other end. Packet loss over a path is the difference between the number of packets transmitted at the starting interface of the path and received at the ending interface of this path.

### 3.1.1. Passive Measurement Method

In order to count the number of packets sent and received and to compare two counters, it is required that the two counters refer exactly to the same set of packets. One difficulty is it is hard to determine exactly when to read the counter since a flow is continuous. A possible solution is to virtually split the flow in consecutive blocks by periodically inserting a delimiting packet, so that each counter refers exactly to the same block of packets. However, delimiting the flow using specific packets requires generating additional packets within the flow and requires the equipment to be able to process those packets. In addition, the method is vulnerable to out of order reception and the loss of delimiting packets.

An existing method by "coloring" IP packets for performance measurement is introduced in [I-D.temple-opsawg-p3m]. This "colored" based approach doesn't use delimiting packets. Instead, it "colours" the packets so that the packets belonging to the same block will have the same "colour", while consecutive blocks will have different colours. Each change of colour represents a sort of auto-synchronization signal that guarantees the consistency of measurements taken by different devices along the path.

### 3.2. One-way Delay

IPPM has defined a protocol for active one way delay measurement OWAMP in [RFC4656] It consists of a control protocol for negotiating measurement sessions and a data plane protocol for test packets. OWAMP is an active protocol meaning that the one delay is measured for artificial packets that are generated for this purpose.

It would be natural to pursue passive and/or hybrid approaches for measuring one way delay. In this case, the goal would be to measure one way delay for packets that are flowing through the network. This can be achieved by defining two observation points that will record the packets they see and the corresponding timestamps. This information will be used to determine the one way delay of the observed packets, similarly than in the active measurement approach. In order to do that, it is necessary to identify which packets are the ones that the measurement will be performed with. One way to do this is to define a certain flow of packets and then record some fields of the packets that are unlikely to change during its journey through their journey between the observation points. Once the packets have been properly identified, and the timestamp information about them has been recorded in the observation points, it should be possible to calculate the one way delay for the observed packets.

If defining a passive metric for one way delay is deemed interesting, it would be then needed to perform a gap analysis for the additional protocols that are needed for this. As the passive approach would also need to negotiate measurement sessions, it may be worth exploring the re use of OWAMP for this. Similarly, both observation points should agree what packet flow will be used for the measurement, so additional negotiation is needed. Finally, IPFIX could be used to report the results so that the actual delay can be calculated.

An additional exercise that would be then relevant is to understand how comparable are measurements obtained through the active and passive measurements. In particular, depending on the packet frequency, it may or may not be possible to achieve the different packet streams available in active measurements.

A hybrid approach for measuring one way delay seems attractive as it would be possible to measure reliably one way delay reusing the packets available in the network when they exist and generating artificial traffic when they don't exist. This requires careful consideration in order to obtain the desired packet streams and it is likely to require additional control protocol to specific the hybrid measurement.

### 3.3. Round-trip Delay

Round-trip delay is used to measure the expected time for network interaction between two hosts on a network; conceptually, it is equivalent to Delay in each direction between the two hosts.

Active measurement of round-trip delay as defined in [RFC2681] requires the observation of test packets transmitted in both directions between two endpoints across a network, a "source" host, which sends the first packet, and a "destination" host, which receives the first test packet and sends a test packet back to the source in reply. The round-trip delay is then calculated as the difference between the time at which the reply is received at the source and the time at which the original test packet was sent from this same source.

IPPM has defined the Two-Way Active Measurement Protocol (TWAMP) [RFC5357] for round trip delay measurement. TWAMP is essentially an extension of OWAMP for the IPPM round-trip delay metric. Like OWAMP, TWAMP consists of a control protocol to negotiate active performance measurement sessions, and a test protocol for transmission of actual test packets.

TWAMP is defined for active performance metrics, which means that the Round-trip delay is measured for packets that are generated specifically for this purpose.

#### 3.3.1. Passive Measurement Method

The passive approach for measuring Round-trip delay would consist on measuring this delay for existent packets in contrast with the active approach in which test packets are generated. Similarly to the method used for measuring One-way Delay, for Round-trip Delay it would be needed to define two observation points that will record the packets they see and the corresponding timestamps.

The procedure for passive measurement of round-trip delay is similar to the procedure for active measurement: a packet sent from a source to a destination is recorded; that packet causes the destination to send a reply back to the source. This reply is also recorded. The packets are identifiable at the source in order to correlate each packet of the round trip in order to calculate a delay.

There are two potential architectures here; one utilizing a source Observation Point (OP) placed topologically close to the source of traffic, and one utilizing an additional destination OP placed topologically close to the destination of traffic.

In order to be able to measure the Round-trip Delay of the observed packets, it would be necessary to identify which packets will be used to perform the measurement.

### 4. Methodology

For certain performance metrics, many passive measurement methodologies may exist. This section gives the functional requirements and design considerations of the passive measurement methodology.

#### 4.1. Measurement Session Management

A measurement session refers to the period of time in which measurement for certain performance metrics is enabled over a forwarding path. When an interface on the measurement node is activated, the interfaces start collecting statistics. When both the upstream and downstream measurement interfaces are activated, the measurement session starts. During a measurement session, data from two active interfaces are periodically collected and the performance metrics, such as loss rate or delay, are derived. A measurement session SHOULD be started either proactively or on demand.

#### 4.1.1. Measurement Configuration

A measurement session can be configured statically. In this case, network operators activate the two interfaces or configure their parameter settings on the relevant nodes either manually or automatically through agents of network management system (NMS). Alternatively, a measurement session can be configured dynamically. In this case, an interface may coordinate another interface on its forwarding path to start or stop a session. Accordingly, the format and process routines of the measurement session control packets need to be specified. The delivery of such packets SHOULD be reliable and it MUST be possible to secure the delivery of such packets.

#### 4.2. Measurement Result Report

Performance reports contain streams of measurement data over a period of time. A data collection agent MAY actively poll the monitoring nodes and collect the measurement reports from all active interfaces. Alternatively, the monitoring nodes might be configured to upload the reports to the specific data collection agents once the data become available. To save bandwidth, the content of the reports might be aggregated and compressed. The period of reporting SHOULD be able to be configured or controlled by rate limitation mechanisms.

#### 4.3. Synchronization

During a measurement session, data from the active upstream and downstream interfaces are periodically collected and the performance metrics are derived. Certain synchronization mechanism is required to ensure the data are correlated. This may further requires that the upstream and downstream interfaces having a certain time synchronization capability (e.g., supporting the Network Time Protocol (NTP) [RFC5905], or the IEEE 1588 Precision Time Protocol (PTP) [IEEE1588].) For packet delay measurement, this requirement for time synchronization is already present.

#### 4.4. Scalability

The measurement methodology MUST be scalable. A service provider production network usually comprises of thousands of nodes. Given the scale, the collecting, processing and reporting overhead of performance measurement data SHOULD NOT overwhelm either monitoring nodes or management nodes. The volume of reporting traffic should be reasonable and not cause any network congestion.

#### 4.5. Robustness

The measurements MUST be independent of the failure of the underlying network. For example, the correct measurement result SHOULD be generated even if some measurement coordinating packets are lost; invalid performance reports should be able to be identified in case that the underlying network is undergoing drastic changes. If dynamic measurement configuration is supported, the delivery of measurement session control packets SHOULD be reliable so that the measurement sessions can be started, ended and performed in a predictable manner.

#### 4.6. Security

The measurement methodology MUST not impose security risks on the network. For example, the monitoring nodes should be prevented from being exploited by third parties to control measurement sessions arbitrarily, which might make the nodes vulnerable for DDoS attacks. If dynamic configuration is supported, the measurement session control packets need to be encrypted and authenticated.

### 5. Security Considerations

[EDITOR'S NOTE: this section will discuss general security considerations of using passive measurement for performance, both on the potential for attacks against the measurement system as well as the potential for privacy or security threats posed by the measurement system itself.]

### 6. IANA Considerations

This document contains no considerations for IANA.

### 7. References

#### 7.1. Normative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.

#### 7.2. Informative References

- [I-D.tempia-opsawg-p3m] Capello, A., Cociglio, M., Castaldelli, L., and A. Bonda, "A packet based method for passive performance monitoring", draft-tempia-opsawg-p3m-04 (work in progress), February 2014.

- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

#### Authors' Addresses

Brian Trammell  
Swiss Federal Institute of Technology Zurich  
Gloriastrasse 35  
8092 Zurich  
Switzerland

Phone: +41 44 632 70 13  
Email: trammell@tik.ee.ethz.ch

Lianshu Zheng  
Huawei Technologies  
China

Email: vero.zheng@huawei.com

Sofia Silva  
LACNIC  
Uruguay

Email: sofia@lacnic.net



Marcelo Bagnulo  
Universidad Carlos III de Madrid  
Av. Universidad 30  
Leganes  
Spain

Email: [bagnulo@it.uc3m.es](mailto:bagnulo@it.uc3m.es)