

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

M. Chen
H. Liu
Y. Yin
R. Papneja
Huawei Technologies
S. Abhyankar
Vodafone
G. Deng
CNNIC
Y. Huang
China Unicom
October 21, 2013

Coloring based IP Flow Performance Measurement Framework
draft-chen-ippm-coloring-based-ipfpm-framework-01

Abstract

By changing one or more bits of packets to "color" the packets into different color blocks, it naturally gives a way to measure the real packet loss and delay without inserting any extra OAM packets. This is called "coloring" based IP Flow Performance Measurement (IPFPM). This document specifies a framework for this "coloring" based IPFPM and defines a new application to the IPFIX for exporting the performance measurement statistic data.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Overview and Concept	4
3. Reference Model and Functional Components	5
3.1. Reference Model	5
3.2. Functional Components	6
3.2.1. Measurement Control Point	6
3.2.2. Data Collecting Point	7
3.2.3. Target Logical Port	8
4. Principles of Coloring based IPFPM	8
4.1. Packet Loss Measurement	8
4.2. Packet Delay Measurement	9
5. Consideration on Color Bits Selection	10
6. Statistic Information Report	11
6.1. IPFIX for Coloring based IPFPM	12
6.1.1. Information Element for IPFPM	12
6.1.2. Templates for IPFPM	14
7. IANA Considerations	20
8. Security Considerations	20
9. Acknowledgements	21
10. References	21
10.1. Normative References	21
10.2. Informative References	21
Authors' Addresses	22

1. Introduction

Performance Measurement (PM) is an important tool that can not only provide Service Level Agreement (SLA) verification but facilitate in troubleshooting (e.g., fault localization or fault delimitation) and network visualization.

There are two typical types of performance measurement: one is active performance measurement, and the other is passive performance measurement.

In active performance measurement the receiver measures the injected packets to evaluate the performance of a path. The active measurement measures the performance of the extra injected packets, the rate, numbers and interval of the injected packets will largely affect the accuracy of the results. In addition, it also requires that the injected packets have to follow the same path as the real traffic; this normally cannot be guaranteed in the pure IP network. The One-Way Active Measurement Protocol (OWAMP) [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) [RFC5357] are tools to enable active performance measurement.

In passive performance measurement, no artificial traffic is injected into the flow and measurements are taken to record the performance metrics of the real traffic. The Multiprotocol Label Switching (MPLS) PM protocol [RFC6374] for packet loss is an example of passive performance measurement. By periodically inserting auxiliary Operations, Administration and Maintenance (OAM) packets, the traffic is delimited by the OAM packets into consecutive blocks, and the receivers count the packets and calculate the packets loss each block.

But, when the OAM channel is in-band, solutions like [RFC6374] are not pure passive measurement as the OAM packets are inserted into the data stream. Furthermore because solutions like [RFC6374] depend on the fixed positions of the delimiting OAM packets for packets counting, they are vulnerable to out-of-order arrival of packets. This could happen particularly with out-of-band OAM channels, but might also happen with in-band OAM because of the presence of multipath forwarding within the network. Out of order delivery of data and the delimiting OAM can give rise to inaccuracies in the performance measurement figures. The scale of these inaccuracies will depend on data speeds and the variation in delivery, but with out-of-band OAM, this could result in significant differences between real and reported performance.

This document describes a mechanism where data packets are marked or "colored" so that they form blocks of data. No additional delimiting

OAM is needed and the performance can be measured in-service without the insertion of additional traffic. Furthermore, because coloring based IP performance measurement does not require extra OAM packets for traffic delimitation, it can be used in situations where there is packets re-ordering. This document specifies a framework for the "coloring" based IP Flow Performance Measurement (IPFPM). This document also defines a new application of and some extensions to the IP Flow Information eXport (IPFIX) for exporting the performance measurement statistic data.

2. Overview and Concept

The concept of "coloring" IP packets for performance measurement is described in [I-D.templia-opsawg-p3m]. Coloring of packets in a specific IP flow to different colors divides the flows into different consecutive blocks. Packets in a block have same color and consecutive blocks will have different colors. This enables the measuring node to count and calculate packet loss and/or delay based for each color block without any additional auxiliary OAM packets. The following figure (Figure 1) is an example that illustrates the different colors in a single IP flow in interleaved red and blue blocks.

```
| Red Block | Blue Block | Red Block | Blue Block |  
RRRRRRRRRRR BBBBBBBBBBBB RRRRRRRRRRRR BBBBBBBBBBBB
```

Figure 1: Packet Coloring

For packet loss measurement, there are two ways to color packets: fixed packet numbers or fixed time period for each color block. This document considers only fixed time period. The sender and receiver nodes count the transmitted and received packets/octets based on each color block. By counting and comparing the transmitted and received packets/octets, the packet loss can be detected.

For packet delay measurement, there are two solutions. One is similar to the packet loss, that it still colors the IP flows to different color blocks and uses the time of the color change as the reference time for delay calculations. This solution requires that there must not be any out-of-order packets; otherwise, the result will not be accurate. Because it uses the first packet of each color block for delay measurement, if there is packet reordering, the first packet of each block at the sender will be probably different from the first packet of the block at the receiver. The alternate way is to periodically coloring a single packet in the IP flow. Within a given time period, there is only one packet that can be colored. The

sender records the timestamp when the colored packet is transmitted, the receiver records the timestamp when detecting the colored packet. With the two timestamps, the packet delay can be computed.

To make the above solutions work, two preconditions are required. The first is that there has to be a way to collect the packet counts and timestamps from the senders and receivers to a centralized calculation element. The IP Flow Information eXport (IPFIX) [RFC5101] protocol is used for collecting the performance measurement statistic information (Section 5 of this document). The second is that the centralized calculation element has to know exactly what packet pair counts (one from the sender and the other is from the receiver) are based on the same color block and a pair of timestamps (one from the sender and the other is from the receiver) are based on the same colored packet. The "Period Number" based solution (Section 4.2 of this document) is introduced to achieve this.

3. Reference Model and Functional Components

3.1. Reference Model

A Multipoint-to-Multipoint (MP2MP) reference model (as shown in Figure 2) is introduced in this document. For a specific IP flow, there may be one or more upstream and downstream Measurement Points (MPs). An IP flow can be identified by the Source IP (SIP) and Destination IP (DIP) addresses, and it may combine the SIP and DIP with any or all of the Protocol number, the Source port, the Destination port, and the Type of Service (TOS) to identify an IP flow. For each flow, there will be a flow identifier that is unique within a certain administrative domain.

An MP is a network node. From the measurement point of view, it consists of two parts (as shown in Figure 3): Data Collecting Point (DCP), and Target Logical Port (TLP). For an MP, there is only one DCP and may be one or more TLPs. The Measurement Control Point (MCP) is a centralized calculation element, MPs periodically report their measurement data to the MCP for final performance calculation. The report protocol is defined in Section 5 of this document.

The reason for choosing MP2MP model is that it can satisfy all the scenarios that include Point-to-Point (P2P), Point-to-Multipoint (P2MP), Multipoint-to-Point (MP2P), and MP2MP. P2P scenario is obvious and can be used anywhere. P2MP and MP2P are very common in mobile backhaul networks. For example, a Cell Site Gateway (CSG) multi-homing to two Radio Network Controller (RNC) Site Gateways (RSGs) is a typical network design. When there is a failure, there is a requirement to monitor the flows between the CSG and the two RSGs hence to determine whether the fault is in the transport network

or in the wireless network (typically called "fault delimitation"). This is especially useful in the situation where the transport network belongs to one service provider and wireless network belongs to other service providers.

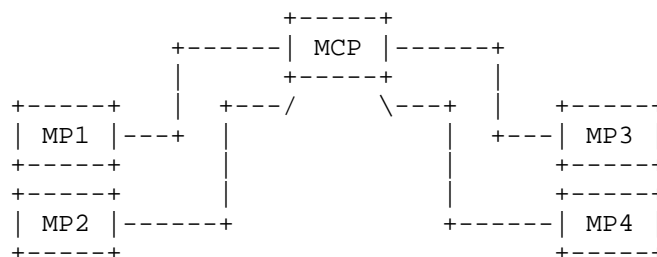


Figure 2: MP2MP based Model

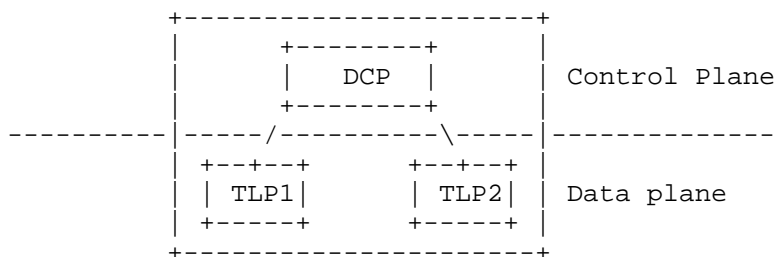


Figure 3: Measurement Point

3.2. Functional Components

3.2.1. Measurement Control Point

The MCP is responsible for calculating the final performance metrics according to the received measurement data from the MPs (actually from the DCPs). For packet loss, based on each color block, the difference between the total counts received from all upstream MPs and the total counts received from all downstream MPs are the lost packet numbers. The MCP must make sure that the counts from the upstream MPs and downstream MPs are related to the same color/packets block. For packet delay (e.g., one way delay), the difference between the timestamps from the downstream MP and upstream MP is the packet delay. Similarly to packet loss, the MCP must make sure the two timestamps are based on the same colored packet.

This document introduces a Period Number (PN) based synchronization mechanism to help the MCP to determine whether any two or more packet counts (from distributed MPs) are related to the same color block or

any two timestamps are related to the same colored packet. The PN is generated each time a DCP reads the packet counts or timestamps from the TLP, and is equal to the modulo of the local time (when the counts or timestamps are read) and the interval of the color time period. Each packet count and timestamp has a PN when reported to the MCP, and the same PN means that they are related to the same color block or colored packet. This requires that the upstream and downstream MPs having a certain time synchronization capability (e.g., supporting the Network Time Protocol (NTP) [RFC5905], or the IEEE 1588 Precision Time Protocol (PTP) [IEEE1588].) and assumes that the upstream and downstream MPs have already time synchronized. Since there is the intention to measure packet delay, this requirement for time synchronization is already present.

3.2.2. Data Collecting Point

The DCP is responsible for periodically collecting the measurement data from the TLPs and reporting the data to the MCP. In addition, when to change the color, when to color a packet (for packet delay measurement), and when to read the packet counts and timestamps are also controlled by DCP. Each DCP will maintain two timers, one (C-timer, used at upstream DCP) is for color changing, the other (R-timer, used at downstream DCP) is for reading the packet counts and timestamps. The two timers have the same time interval but are started at different times. A DCP can be either an upstream or a downstream DCP: the role is specific to an IP flow. For a specific IP flow, the upstream DCP will change the color and read the packet counts and timestamps when the C-timer expires, the downstream DCP just reads the packets counts and timestamps when the R-timer expires.

In order to allow for a certain degree of packets re-ordering, the R-timer should be started later than a defined period of time (Δt) after the C-timer is started, so as long as the Δt satisfies the following conditions:

$$(\text{Time-L} - \text{Time-MRO}) < \Delta t < (\text{Time-L} + \text{Time-MRO})$$

where:

Time-L: the link delay time between the sender and receiver;

Time-MRO: the maximum re-ordering time difference; if a packet is expected to arrive at t_1 but actually arrives at t_2 , then the Time-MRO = $|t_2 - t_1|$.

So, the R-timer should be started at " $t + \Delta t$ " (where the t is the time when C-timer started).

For simplicity, the C-timer should be started at the beginning of each time period. This document recommends the implementation to support at least these time periods (1s, 10s, 1min, 10min and 1h). So, if the time period is 10s, the C-timer should be started at the time of any multiples of 10 in seconds (e.g., 0s, 10s, 20s, etc.), then the R-timer should be started (e.g., 0s+ Δt , 10s+ Δt , 20s+ Δt , etc.). With this method, each DCP can independently start its C-timer and R-timer given that the clocks have been synchronized.

3.2.3. Target Logical Port

The TLP is a logical entity that actually executes the final measurement actions (e.g., colors the packets, counts the packets, records the timestamps, etc.). Normally, a physical interface corresponds to a TLP, and the TLP resides in the data plane. For a measurement instance (corresponding to an IP flow), a TLP will maintain a pairs of packet counters and a timestamp counter for each color block. As for the pair of packet counters, one is for counting packets and the other is for counting octets.

4. Principles of Coloring based IPFPM

To simplify the process description, the flows discussed in this document are all unidirectional. A bidirectional flow can be seen as two unidirectional flows. For a specific flow, there will be upstream and downstream TLPs and upstream and downstream packet counts/timestamp accordingly.

4.1. Packet Loss Measurement

For packet loss measurement, this document defines the following counters and quantities:

U-CountP[n][m]: U-CountP is a two-dimensional array that stores the number of packets transmitted by each upstream TLP in each color time period. Specifically, parameter "n" is the "period number" of measured color blocks while parameter "m" refers to the m-th TLP of the upstream TLPs.

D-CountP[n][m]: D-CountP is a two-dimensional array that stores the number of packets received by each downstream TLP in each color time period. Specifically, parameter "n" is the "period number" of measured color blocks while parameter "m" refers to the m-th TLP of the downstream TLPs.

U-CountO[n][m]: U-CountO is a two-dimensional array that stores the number of octets transmitted by each upstream TLP in each color time

period. Specifically, parameter "n" is the "period number" of measured color blocks while parameter "m" refers to the m-th TLP of the upstream TLPs.

D-CountO[n][m]: D-CountO is a two-dimensional array that stores the number of octets received by each downstream TLP in each color time period. Specifically, parameter "n" is the "period number" of measured color blocks while parameter "m" refers to the m-th TLP of the downstream TLPs.

LossP: the number of packets transmitted by the upstream TLPs but not received at the downstream TLPs.

LossO: the total octets transmitted by the upstream TLPs but not received at the downstream TLPs.

The total packet loss of a flow can be computed as follows:

$$\text{LossP} = \text{U-CountP}[1][1] + \text{U-CountP}[1][2] + \dots + \text{U-CountP}[n][m] - \text{D-CountP}[1][1] - \text{D-CountP}[1][2] - \dots - \text{D-CountP}[n][m'].$$

$$\text{LossO} = \text{U-CountO}[1][1] + \text{U-CountO}[1][2] + \dots + \text{U-CountO}[n][m] - \text{D-CountO}[1][1] - \text{D-CountO}[1][2] - \dots - \text{D-CountO}[n][m'].$$

Where the m and m' are the number of upstream TLPs and downstream TLPs, respectively.

4.2. Packet Delay Measurement

For packet delay measurement, there will be only one upstream TLP and may be one or more (P2MP) downstream TLPs. Although the coloring based IPFPM supports P2MP model, this document only discusses P2P model, the P2MP model is left for future study. This document defines the following timestamps and quantities:

U-Time[n]: U-Time is a one-dimension array that stores the time when colored packets are sent; parameter "n" is the "period number" of colored packets.

D-Time[n]: D-Time is a one-dimension array that stores the time when colored packets are received; parameter "n" is the "period number" of colored packets. This is only for P2P model.

D-Time[n][m]: D-Time a two-dimension array that stores the time when the colored packet is received by downstream TLPs at each color time period. Here, parameter "n" is the "period number" of colored packets while parameter "m" refers to the m-th TLP of the downstream TLPs. This is for P2MP model which is left for future study.

One-way Delay[n]: The one-way delay metric for packet networks is described in [RFC2679]. The "n" identifies the "period number" of the colored packet.

$$\text{One-way Delay}[1] = \text{D-Time}[1] - \text{U-Time}[1].$$
$$\text{One-way Delay}[2] = \text{D-Time}[2] - \text{U-Time}[2].$$

...

$$\text{One-way Delay}[n] = \text{D-Time}[n] - \text{U-Time}[n].$$

In the case of two-way delay, the delay is the sum of the two one-way delays of the two flows that have the same TLPs but have opposite directions.

$$\text{Two-way Delay}[1] = (\text{D-Time}[1] - \text{U-Time}[1]) + (\text{D-Time}'[1] - \text{U-Time}'[1]).$$
$$\text{Two-way Delay}[2] = (\text{D-Time}[2] - \text{U-Time}[2]) + (\text{D-Time}'[2] - \text{U-Time}'[2]).$$

...

$$\text{Two-way Delay}[n] = (\text{D-Time}[n] - \text{U-Time}[n]) + (\text{D-Time}'[n] - \text{U-Time}'[n]).$$

Where the D-Time and U-Time are for one forward flow, the D-Time' and U-Time' are for reverse flow.

5. Consideration on Color Bits Selection

This document does not specify which bits in IP header should be used for coloring; it primarily introduces options that the operator can choose for packet coloring. This document introduces the following options:

1. For IPv4, there is only one bit (the last reserved bit of the Flag field of the IPv4 header) that can be used for coloring. With one bit, at any time it can only be used for loss or delay measurement and cannot be used for packet loss and delay measurement simultaneously;
2. For IPv6, it can leverage the IPv6 extension header for coloring, for example, adding a new option to the Hop-by-Hop Options header[RFC2460] for coloring. More detail will be added in a future version or in a separate document.

For the above options, the operators should carefully think of the color bits selection to make sure that the setting or changing of the color bits SHOULD NOT affect the normal packet forwarding and process.

The implementations SHOULD provide knobs for operators to configure and change the color bits according to their network design and policies.

6. Statistic Information Report

As described in Section 4 of this document, during the performance measurement period, each DCP periodically reports performance measurement statistic information to the MCP, and the MCP will compute the final performance measurement results according to the received statistic information.

For a specific IP flow, for either packet delay or loss measurement, there will be at least one upstream and one downstream DCP. For accurate measurements, time synchronization is required and the Period Number is used by MCP to uniquely identify and correlate the packet counts/ timestamps between the upstream and downstream DCPs for a specific color block or colored packet.

For packet loss measurement, the following information is required to report to the MCP:

DCP Identifier

TLP Identifier

Flow Identifier

Period Number

Packet Number Count

Packet Octets Count

For packet delay measurement, the following information is required to report to the MCP:

DCP Identifier

TLP Identifier

Flow Identifier

Period Number

Timestamp

In addition, a DCP may report some status (e.g., whether a DCP is time synchronized) to the MCP, hence to help the MCP to determine whether measurement data from a DCP is valid and can be used for computation. The following information may be included:

DCP Identifier

DCP Status

TLP Status

6.1. IPFIX for Coloring based IPFPM

The IPFIX protocol [RFC5101] defines how IP Flow information can be exported from routers, measurement probes, or other devices. It defines many Information Elements [RFC5102] that can be used to carry and export the above information from DCPs to MCP. Except the Period Number, DCP Status and TLP Status, all the above information can be identified with the existing Information Elements.

DCP Identifier: exporterIPv4Address/exporterIPv6Address (130/131)

TLP Identifier: portId (142)

Flow Identifier: flowId (148)

Packet Number Count: packetTotalCount (86)

Packet Octets Count: octetTotalCount (85)

Timestamp: flowStartMicroseconds (154)

6.1.1. Information Element for IPFPM

6.1.1.1. periodNumber

Description: The periodNumber is used to identify a packet count or timestamp that belongs to a specific color block or colored packet. The MCP uses it to determine whether any two or more packet counts (from distributed DCPs) are related to the same color block or any two timestamps are related to the same colored packet. The PN is generated each time a DCP reads the packet counts and timestamp from the TLP, and is equal to the modulo of the local time (when the counts and timestamps are read) and the interval of the color time period.

Abstract Data Type: unsigned32

ElementId: TBD1

Status: current

6.1.1.2. dcpStatus

Description: The dcpStatus is used to carry some status of a DCP (For example, whether a DCP has already time synchronized).

Abstract Data Type: unsigned16

ElementId: TBD2

Status: current

The dcpStatus is defined as follows:

```

+---+---+---+---+---+---+---+---+---+---+
|           Reserved           |T|
+---+---+---+---+---+---+---+---+---+---+

```

One bit (the Time synchronized bit) is defined in this document, it is used to indicate whether a DCP is time synchronized. When the T bit set, the DCP is time synchronized, otherwise, the DCP is not time synchronized. The MCP MUST calculate the results when all related DCPs of a flow are time synchronized, otherwise, the results will not correct.

6.1.1.3. tlpStatus

Description: The tlpStatus is used to carry some status of a TLP.

Abstract Data Type: unsigned8

ElementId: TBD3

Status: current

The dcpStaus is defined as follows:

```

+--+--+--+--+--+--+--+
|   Reserved   |U|
+--+--+--+--+--+--+--+

```

One bit (the Upstream TLP bit) is defined in this document; it is used to indicate whether a TLP is the upstream or downstream TLP for an IP flow. When the U bit set, the TLP is the upstream TLP of the flow; otherwise, the TLP is the downstream TLP of the flow.

6.1.2. Templates for IPFPM

6.1.2.1. DCP Status Options Template

The DCP Status Options Template is used to report the status of a DCP; it SHOULD contain the following Information Elements:

meteringProcessId (scope) [IPFIX-IANA]

DCP Identifier

This is the identifier of a DCP, any of the following Information Elements can be used:

exporterIPv4Address [IPFIX-IANA]

exporterIPv6Address [IPFIX-IANA]

dcpStatus

The dcpStatus is as defined in Section 6.1.1.2 of this document.

The Data Records specified by the DCP Status Options Template SHOULD be exported once the IPFIX session established or when status changed.

An example of the DCP Status Options Template Set is 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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

	Set ID = 3		Length = 24	
+-----+		+-----+		+-----+
	Template ID XXX		Field Count = 3	
+-----+		+-----+		+-----+
	Scope Field Count = 1	0	meteringProcessId = 143	
+-----+		+-----+		+-----+
	Scope 1 Field Length = 4	0	exporterIPv4Address = 130	
+-----+		+-----+		+-----+
	Scope 1 Field Length = 4	0	dcpStatus = TBD3	
+-----+		+-----+		+-----+
	Field Length = 1		Padding	
+-----+		+-----+		+-----+

6.1.2.2. Flow Options Template

The Flow Options Template is used to report all the configured flows on a DCP. It SHOULD include the following the Information Elements:

meteringProcessId (scope) [IPFIX-IANA]

TLP Identifier

The identifier of the TLP that is responsible for measuring the flow; portId is used to carry the TLP Identifier:

portId [IPFIX-IANA]

TLP Name

The name of the TLP that is responsible for measuring the flow; interfaceName is used to carry the TLP Name:

interfaceName [IPFIX-IANA]

tlpStatus [Section 6.1.1.3]

flowId [IPFIX-IANA]

protocolIdentifier [IPFIX-IANA]

Source IP address

The source IP address or prefix of an IP flow, for this address, any of the following Information Elements can be used:

sourceIPv4Address [IPFIX-IANA]

sourceIPv6Address [IPFIX-IANA]

sourceIPv4Prefix [IPFIX-IANA]

sourceIPv6Prefix [IPFIX-IANA]

Source IP prefix length

The source IP prefix length of a prefix, any of the following Information Elements can be used:

sourceIPv4PrefixLength [IPFIX-IANA]

sourceIPv6PrefixLength [IPFIX-IANA]

Source port

The source port of an IP flow, any of the following Information Elements can be used:

udpSourcePort [IPFIX-IANA]

tcpSourcePort [IPFIX-IANA]

Destination IP address

The destination IP address or prefix of an IP flow, for this address, any of the following Information Elements can be used:

destinationIPv4Address [IPFIX-IANA]

destinationIPv6Address [IPFIX-IANA]

destinationIPv4Prefix [IPFIX-IANA]

destinationIPv6Prefix [IPFIX-IANA]

Destination IP prefix length

The destination IP prefix length of a prefix, any of the following Information Elements can be used:

destinationIPv4PrefixLength [IPFIX-IANA]

destinationIPv6PrefixLength [IPFIX-IANA]

Destination port

The destination port of an IP flow, any of the following Information Elements can be used:

udpDestinationPort [IPFIX-IANA]

tcpDestinationPort [IPFIX-IANA]

The Data Records specified by the Flow Options Template SHOULD be exported once the IPFIX session established or when the configured flows changed (e.g., a new flow is added for measurement or a flow deleted to stop the measurement).

An example of the Flow Options Template Set is 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	2	3	4	5	6	7	8	9
Set ID = 3										Length = 52																													
Template ID XXX										Field Count = 10																													
Scope Field Count = 1										0	meteringProcessId=143																												
Scope 1 Field Length = 4										0	portId = 142																												
Field Length = 4										0	interfaceName = 82																												
Field Length = 4										0	tlpStatus = TBD3																												
Field Length = 1										0	flowID = 148																												
Field Length = 4										0	protocolIdentifier = 4																												
Field Length = 1										0	sourceIPv4Address = 8																												
Field Length = 4										0	udpSourcePort = 4																												
Field Length = 2										0	destinationIPv4Address = 4																												
Field Length = 4										0	udpDestinationPort = 4																												
Field Length = 4										0	udpDestinationPort = 4																												
Field Length = 2										Padding																													

6.1.2.3. Packet Loss Template

The Packet Loss Template is used by a DCP to report the packet loss measurement statistic of a flow to the MCP; it SHOULD contain the following Information Elements:

TLP Identifier

This is the identifier of a TLP, portId is used to carry the TLP Identifier:

portId[IPFIX-IANA]

flowId[IPFIX-IANA]

periodNumber [Section 6.1.1.1]

packetTotalCount[IPFIX-IANA]

octetTotalCount[IPFIX-IANA]

An example of the Packet Loss Data Set is as follows:

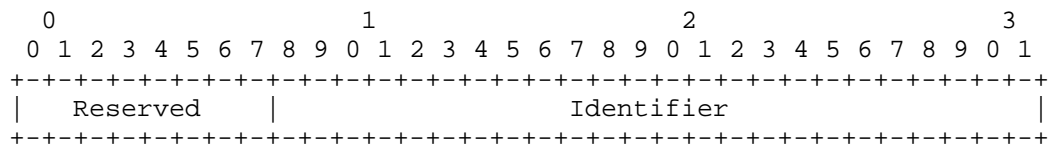
0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 2										Length = 33 octets																													
Template ID XXX										Field Count = 7																													
portId = 142										Field Length = 4																													
flowId = 148										Field Length = 4																													
periodNumber = TBD1										Field Length = 4																													
packetTotalCount = 86										Field Length = 8																													
octetTotalCount = 85										Field Length = 8																													

The portId is used to identify and carry the TLP ID.

The flowId is a identifier that is unique within a specific administrative domain (e.g., an Autonomous System). The TLP, DCP and MCP have to agree a flow identifier related to a specific flow. For

example, the flow identifier can be generated and maintained by a centralized element. How to generate and maintain the flowId is out the scope of this document.

The flowId has the following structure, the Reserved field that is left for future extensions, the Identifier field is 24-bit in length.



The periodNumber is as defined in Section 6.1.1.1 of this document.

The packetTotalCount is used to carry the total transmitted/received packets of a flow since the measurement start.

The octetTotalCount is used to carry the total transmitted/received octets of a flow since the measurement start.

6.1.2.4. Packet Delay Template

The Packet Delay Template is used by a DCP to report the packet delay measurement statistic of a flow to the MCP; it SHOULD contain the following Information Elements:

TLP Identifier

This is the identifier of a TLP, portId is used to carry the TLP Identifier:

portId [IPFIX-IANA]

flowId [IPFIX-IANA]

periodNumber [Section 6.1.1.1]

timestamp

The time when color a packet, flowStartMicroseconds is used to carry the timestamp:

flowStartMicroseconds[IPFIX-IANA]

An example of the Packet Delay Data Set is 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								
Set ID = 2										Length = 25 octets																													
Template ID 258										Field Count = 6																													
0	portId = 142									Field Length = 4																													
0	flowId = 148									Field Length = 4																													
0	periodNumber = TBD1									Field Length = 4																													
0	flowStartMicroseconds = 154									Field Length = 8																													

The flowId is used to carry the flow identifier of a flow; the structure is defined in Section 6.1.2.3 of this document.

The periodNumber is as defined in Section 6.1.1.1 of this document.

The flowStartMicroseconds is used to carry the timestamp of a colored packet of a specific flow.

7. IANA Considerations

The IANA is required to allocate 3 new Information Elements codes for the Information Elements defined in Section 6.1.1 from the IPFIX Information Elements registry.

8. Security Considerations

This document specifies a passive mechanism for measuring packet loss and delay within a Service Provider's network where the IP packets are marked or "colored" with the unused bits in IP head field, and then inserting additional OAM packets during the measurement is avoided. Obviously, such mechanism does not directly affect other applications running on the Internet but may lead to potential affects to the measurement itself.

First, the measurement itself may be affected by routers (or other network devices) along the path of IP packets intentionally altering the value of color bits of packets. Just as mentioned before, the mechanism specified in this document is just in the context of one Service Provider's network, so the routers (or other network devices) are controllable and thus this kind of attack can be omitted.

Second, the measurement can be harmed by attackers injecting artificial traffic. Then authentication techniques, like digital signatures, may be used to guard against such kind of attack.

9. Acknowledgements

The authors would like to thank Adrian Farrel for his review, suggestion and comments to this document.

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.
- [RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.
- [RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.

10.2. 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-03 (work in progress), February 2013.
- [IEEE1588]
IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [IPFIX-IANA]
, "<http://www.iana.org/assignments/ipfix/ipfix.xml>", .
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, April 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. Babiarez, "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.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

Authors' Addresses

Mach(Guoyi) Chen
Huawei Technologies

Email: mach.chen@huawei.com

Hongming Liu
Huawei Technologies

Email: liuhongming@huawei.com

Yuanbin Yin
Huawei Technologies

Email: yinyuanbin@huawei.com

Rajiv Papneja
Huawei Technologies

Email: Rajiv.Papneja@huawei.com

Shailesh Abhyankar
Vodafone
Vodafone House, Ganpat Rao kadam Marg Lower Parel
Mumbai 40003
India

Email: shailesh.abhyankar@vodafone.com

Guangqing Deng
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing
China

Email: dengguangqing@cnnic.cn

Yongliang Huang
China Unicom

Email: huangyl@dipmt.com

IPPM
Internet-Draft
Intended status: Informational
Expires: April 21, 2014

L. Deng
Z. Cao
China Mobile
October 18, 2013

Problem Statement for IP measurement in mobile networks
draft-deng-ippm-wireless-00.txt

Abstract

This document analyzes the potential problems of applying existing IP-based performance measurement methods to wireless accessing environments. It suggests that more robust passive measuring methods and performance metrics 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 April 21, 2014.

Copyright and License Notice

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

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

Table of Contents

1. Introduction	2
2. Motivation	2
2.1. Dynamic Load Balancing	3
2.2. Radio Congestion Detection	4
3. Summary	5
4. Security Considerations	5
5. IANA Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	5
Authors' Addresses	6

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 analyze two usecases the potential problems of applying existing IP-based performance measurement methods to wireless accessing environments, which are tending to utilize resource pooling and dynamic load balancing techniques to accommodate explosively increasing data traffic, and conclude that more robust passive measuring methods and performance metrics are needed.

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 pooled 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.

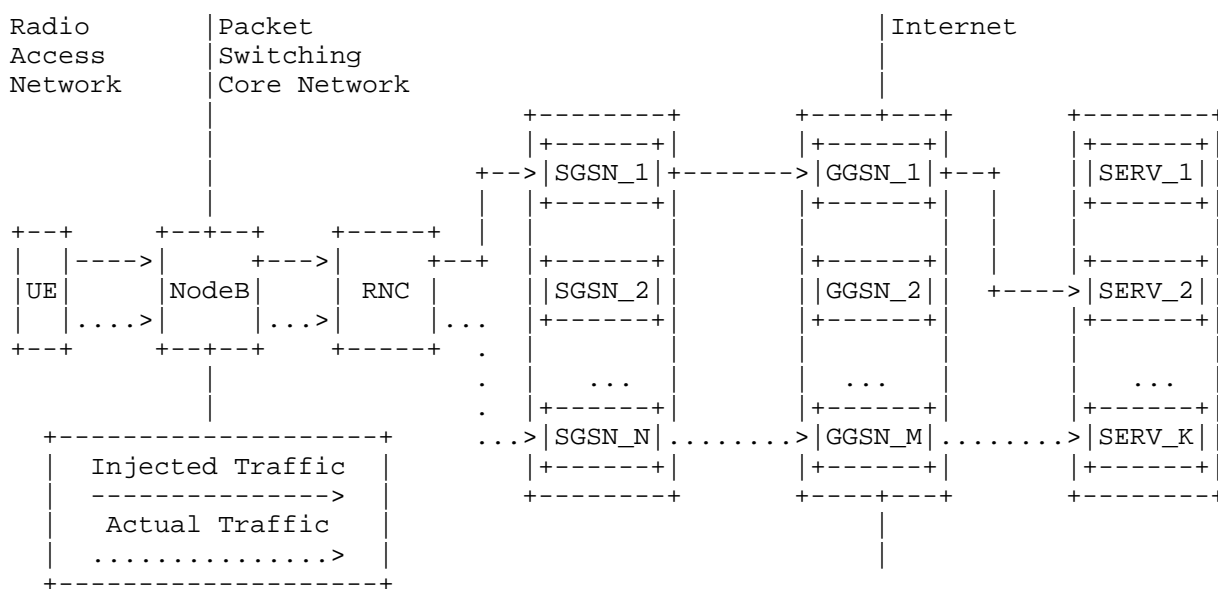


Figure 1: Active Measuring Traffic versus Actual Traffic in case of Device Pooling

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 WiFi 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 active performance measurement methods[I-D.draft-chen-ippm-coloring-based-ipfpm-framework].

3. Summary

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

- o viable passive measurement methodology 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.

4. Security Considerations

TBA

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-00 (work in progress), July 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

Intended Status: Proposed Standard
Expires: April 2014

N. Elkins
B. Jouris
Inside Products
October 16, 2013

IPPM Considerations for the IPv6 PDM Extension Header
draft-elkins-ippm-pdm-metrics-02

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

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

Table of Contents

1	Introduction	3
1.1	Packet Identification Data	3
1.2	Data in the PDM Destination Option Headers	4
2	Metrics Derived from the PDM Destination Options	4
3	Base Derived Metrics	5
3.1	One-Way Delay	5
3.1	Round-Trip Delay	5
3.2	Server Delay	5
4	Sample Implementation Flow (PDM Type 1)	5
4.1	Step 1 (PDM Type 1)	6
4.2	Step 2 (PDM Type 1)	6
4.3	Step 3 (PDM Type 1)	7
4.4	Step 4 (PDM Type 1)	8
4.5	Step 5 (PDM Type 1)	9
5	Sample Implementation Flow (PDM 2)	9
5.1	Step 1 (PDM Type 2)	10
5.2	Step 2 (PDM Type 2)	11
5.3	Step 3 (PDM Type 2)	11
5.4	Step 4 (PDM Type 2)	12
5.5	Step 5 (PDM Type 2)	13
6	Derived Metrics : Advanced	14
6.1	Advanced Derived Metrics : Triage	14
6.2	Advanced Derived Metrics : Network Diagnostics	14
6.2.1	Retransmit Duplication (RD)	15
6.2.2	ACK Lag (AL)	16
6.2.3	Third-party Connection Reset (TPCR)	16
6.2.4	Potential Hang (PH)	16
6.3	Advanced Metrics : Session Classification	16
7	Use Cases	17
8	Security Considerations	17
9	IANA Considerations	17
10	References	18
10.1	Normative References	18
10.2	Informative References	18
11	Acknowledgments	18
	Authors' Addresses	18

1 Introduction

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.

For background, please see draft-ackermann-tictoc-pdm-ntp-usage-00 [ACKPDM], draft-elkins-6man-ipv6-pdm-dest-option-03 [ELKPDM], draft-elkins-v6ops-ipv6-packet-sequence-needed-01 [ELKPSN], draft-elkins-v6ops-ipv6-pdm-recommended-usage-01 [ELKUSE], and draft-elkins-v6ops-ipv6-end-to-end-rt-needed-01 [ELKRSP]. These drafts are companions to this document.

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' data. That is, they are of the traffic actually traveling on the network.

1.1 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.)

1.2 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.

2 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.

3 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

3.1 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].

3.1 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].

3.2 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.

4 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. The descriptions of these fields is in draft-elkins-6man-ipv6-pdm-dest-option-02 [ELKPDM].

Time synchronization is required between Host A and Host B. See draft-ackermann-tictoc-pdm-ntp-usage-00 [ACKPDM] 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.

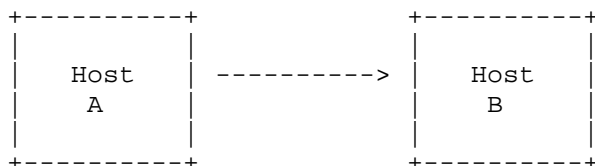
4.1 Step 1 (PDM Type 1)

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.

4.2 Step 2 (PDM Type 1)

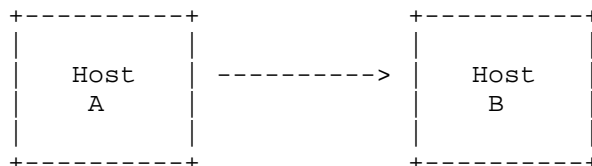
Packet 1 is received by Host B. The time for Host B was synchronized with Host A. Both were set initially to 10:00AM.

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

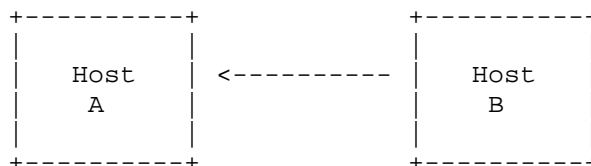
If we make the substitutions from our sample case above, then:

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

4.3 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.

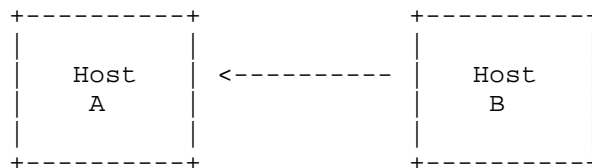
4.4 Step 4 (PDM Type 1)

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

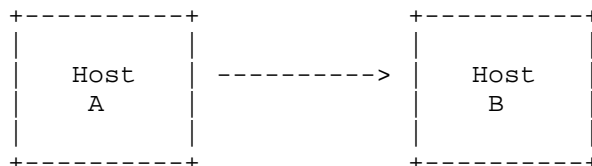
If we make the substitutions from our sample case above, then:

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

4.5 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.

5 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, 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.

5.1 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.

5.2 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.

5.3 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:

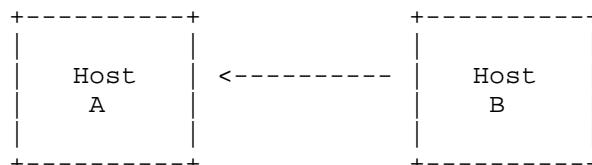
```
PSNLR : Packet Sequence Number Last Received:    25
TSLR  : Timestamp Last Received                  :    11:00:03
PSNTP : Packet Sequence Number This Packet      :    12
TSTP  : Timestamp This Packet                    :    11:00:07
```

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.

5.4 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:

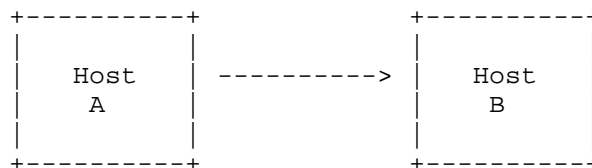
Two-way delay = 12 - 4 or 8

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.

5.5 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

6 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.

6.1 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.

6.2 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.

6.2.1 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.

6.2.2 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.

6.2.3 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.

6.2.4 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.

6.3 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.

7 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

8 Security Considerations

There are no security considerations.

9 IANA Considerations

There are no IANA considerations.

10 References

10.1 Normative References

- [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.

10.2 Informative References

- [ACKPDM] Ackermann, M., "draft-ackermann-tictoc-pdm-ntp-usage-00", Internet Draft, September 2013.
- [ELKPDM] Elkins, N., "draft-elkins-6man-ipv6-pdm-dest-option-04", Internet Draft, October 2013.
- [ELKPSN] Elkins, N., "draft-elkins-v6ops-ipv6-packet-sequence-needed-01", Internet Draft, September 2013.
- [ELKRSP] Elkins, N., "draft-elkins-v6ops-ipv6-end-to-end-rt-needed-01", Internet Draft, September 2013.
- [ELKUSE] Elkins, N., "draft-elkins-v6ops-ipv6-pdm-recommended-usage-01", Internet Draft, September 2013

11 Acknowledgments

The authors would like to thank Al Morton, 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
<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
<http://www.insidethestack.com>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2014

J. Hedin
G. Mirsky
S. Baillargeon
Ericsson
October 2, 2013

Differentiated Service Code Point and Explicit Congestion Notification
Monitoring in Two-Way Active Measurement Protocol (TWAMP)
draft-hedin-ippm-type-p-monitor-02

Abstract

This document describes an OPTIONAL feature for TWAMP [RFC5357] allowing the monitoring of the Differentiated Service Code Point and Explicit Congestion Notification fields with the TWAMP-Test protocol.

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 5, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Conventions used in this document	3
1.1.1. Terminology	3
1.1.2. Requirements Language	3
2. TWAMP Extensions	4
2.1. Setting Up Connection to Monitor DSCP and ECN	4
2.2. TWAMP-Test Extension	4
2.2.1. Session-Reflector Packet Format for DSCP and ECN Monitoring	4
2.2.2. DSCP and ECN Monitoring with RFC 6038 extensions	7
2.2.3. Consideration for TWAMP Light mode	8
3. IANA Considerations	8
4. Security Considerations	9
5. Acknowledgements	9
6. References	9
6.1. Normative References	9
6.2. Informative References	10
Authors' Addresses	10

1. Introduction

One-Way Active Measurement Protocol (OWAMP) [RFC4656] defines Type-P descriptor and negotiation of its value in OWAMP-Control protocol. Two-Way Active Measurement Protocol (TWAMP) [RFC5357] states that only Differentiated Service Code Point (DSCP) value can be defined by Type-P descriptor and the negotiated value must be used by both Session-Sender and Session-Reflector. The TWAMP specification also states that the same value of DSCP (found in the Session-Sender packet) MUST be used in the test packet reflected by the Session-Reflector. However the TWAMP-Test protocol does not specify any methods to determine or report when the DSCP value has changed or is different than expected in the forward or reverse direction. Re-marking the DSCP (changing its original value) in IP networks is possible and often accomplished by a Diffserv policy configured on a single node along the IP path. In many cases, a change of the DSCP value of indicates an unintentional or erroneous behavior. At best, the Session-Sender can detect a change of the DSCP reverse direction assuming such change is actually detectable.

This document describes an OPTIONAL feature for TWAMP. It is called the DSCP and ECN monitoring feature. This feature allows the Session-Sender to know the actual DSCP value received at the Session-Reflector. Furthermore this OPTIONAL feature also tracks the Explicit Congestion Notification (ECN) value received at the Session-Reflector. This is helpful to determine if ECN is actually operating or if an ECN-capable node has detected congestion in the forward direction.

1.1. Conventions used in this document

1.1.1. Terminology

DSCP: Differentiated Service Codepoint

ECN: Explicit Congestion Notification

IPPM: IP Performance Measurement

TWAMP: Two-Way Active Measurement Protocol

OWAMP: One-Way Active Measurement Protocol

1.1.2. Requirements Language

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].

2. TWAMP Extensions

TWAMP connection establishment follows the procedure defined in Section 3.1 of [RFC4656] and Section 3.1 of [RFC5357] where the Modes field been used to identify and select specific communication capabilities. At the same time the Modes field been recognized and used as extension mechanism [RFC6038]. The new feature requires new bit position to identify the ability of a Session-Reflector to return value of received DSCP and ECN values back to a Session-Sender, and to support the new Session-Reflector packet format in the TWAMP-Test protocol. See the Section 3 for details on the assigned value and bit position.

2.1. Setting Up Connection to Monitor DSCP and ECN

The Server sets DSCP and ECN Monitoring flag in Modes field of the Server Greeting message to indicate its capabilities and willingness to monitor them. If the Control-Client agrees to monitor DSCP and ECN on some or all test sessions invoked with this control connection, it MUST set the DSCP and ECN Monitoring flag in Modes field in the Setup Response message.

2.2. TWAMP-Test Extension

Monitoring of DSCP and ECN requires support by Session-Reflector and changes format of its test packet format both in unauthenticated, authenticated and encrypted modes. Monitoring of DSCP and ECN does not alter Session-Sender test packet format but certain considerations must be taken when and if this mode is accepted in combination with Symmetrical Size mode[RFC6038].

2.2.1. Session-Reflector Packet Format for DSCP and ECN Monitoring

When Session-Reflector supports DSCP and ECN Monitoring it MUST construct Sender DSCP and ECN (S-DSCP-ECN) field for each test packet it sends to Session-Sender according to the following procedure:

- first six bits MUST be copied Differentiated Service field from received Session-Sender test packet into Sender DSCP (S-DSCP) field;
- following two bits MUST be copied ECN field from received Session-Sender test packet into Sender ECN (S-ECN) field.

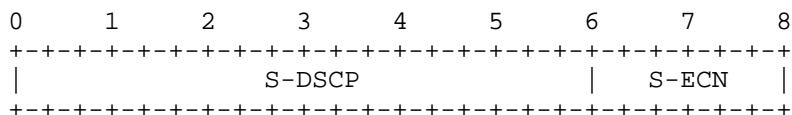


Figure 1: Sender DSCP and ECN field format

For unauthenticated mode:

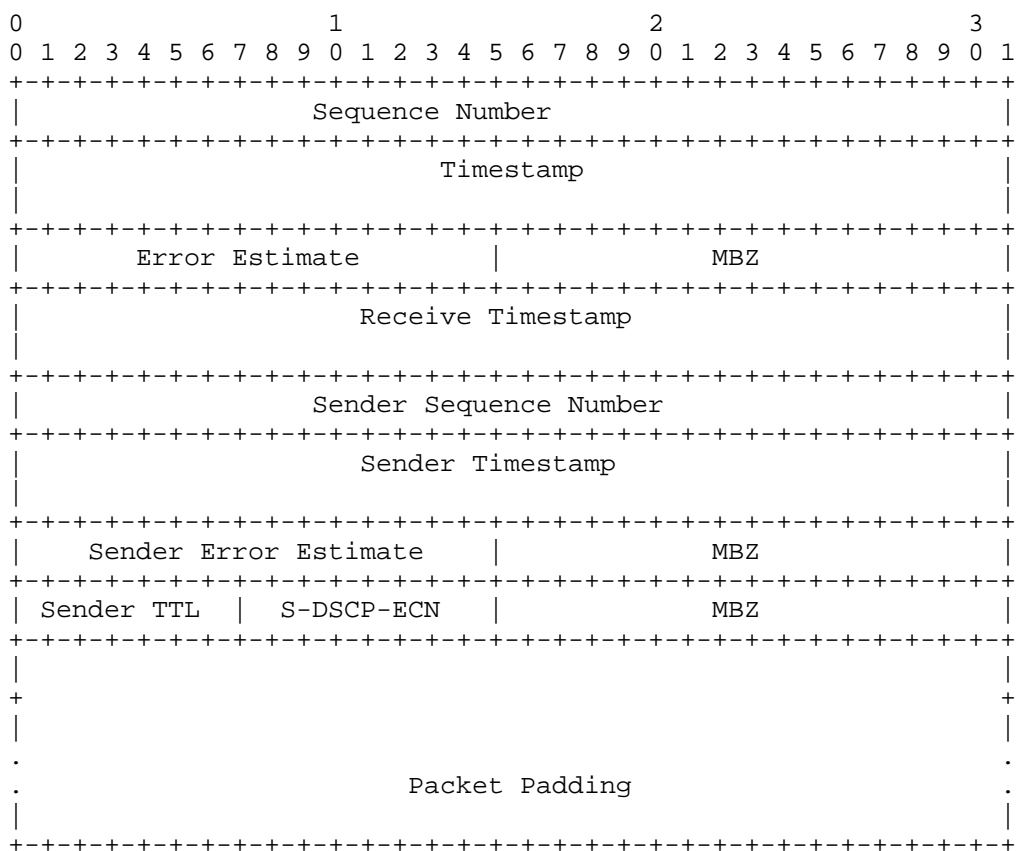
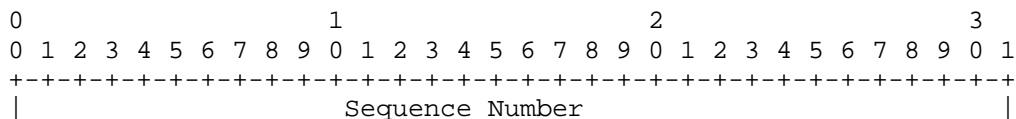


Figure 2: Session-Reflector test packet format with DSCP and ECN monitoring in unauthenticated mode

For authenticated and encrypted modes:



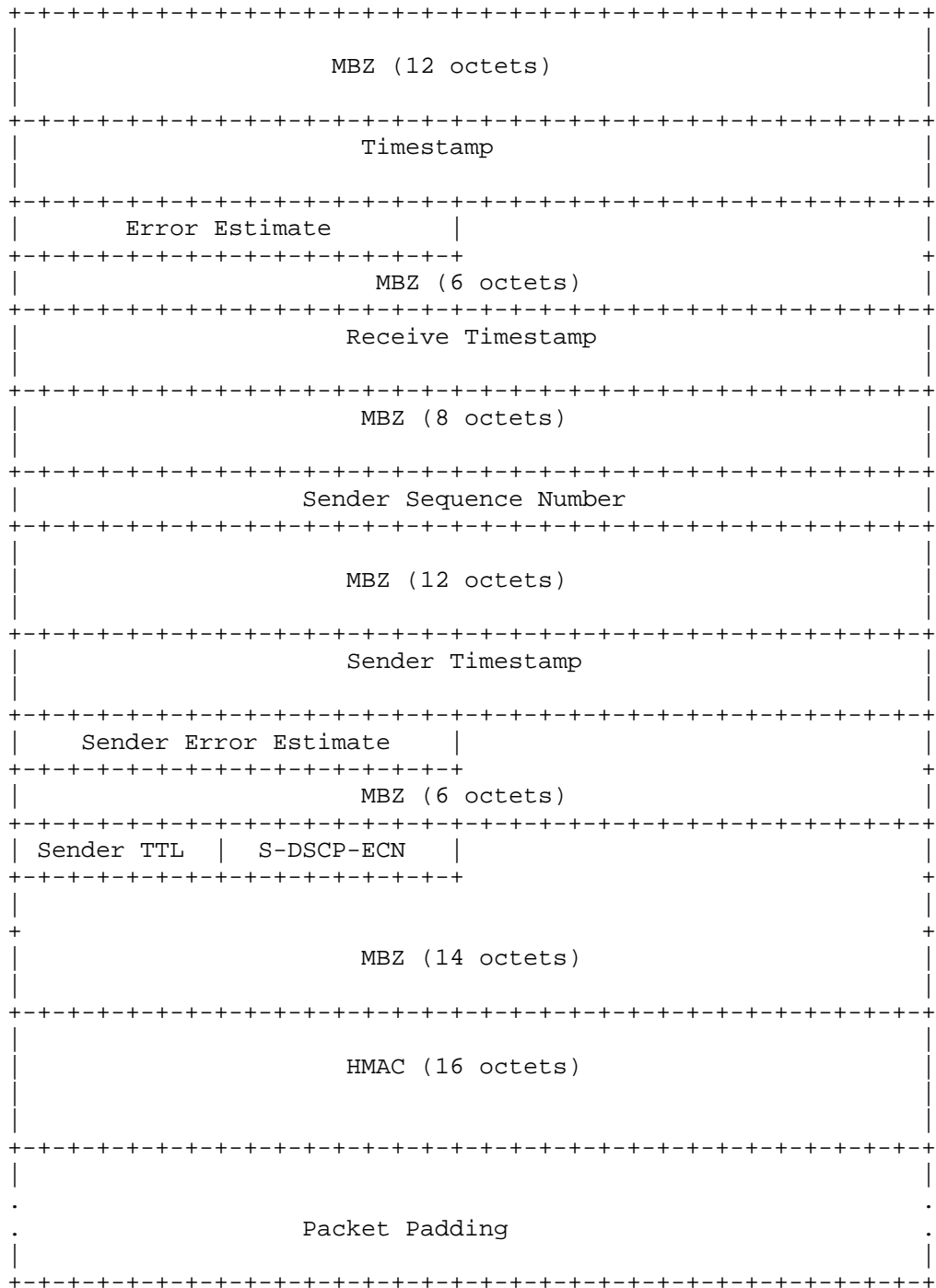


Figure 3: Session-Reflector test packet format with DSCP and ECN monitoring in authenticated or encrypted modes

The DSCP value is often copied into reflected test packets with current TWAMP implementations (with or without TWAMP-Control protocol). With DSCP and ECN Monitoring Extension Session-Reflector handles DSCP as following:

The Session-Reflector MUST extract the S-DSCP-ECN value from the DSCP and ECN values of received packets;

The Session-Reflector MUST transmit each reflected test packet with DSCP set to the negotiated/provisioned value;

If the negotiated/provisioned DSCP value is not known (e.g. TWAMP Light), the choice of the DSCP is implementation specific. For instance, Session-Reflector MAY copy the DSCP value from the received test packet and set it as DSCP in a reflected packet.

2.2.2. DSCP and ECN Monitoring with RFC 6038 extensions

[RFC6038] defined two extensions to TWAMP. First, to ensure that Session-Sender and Session-Reflector exchange TWAMP-Test packets of equal size. Second, to specify number of octets to be reflected by Session-Reflector. If DSCP and ECN monitoring and Symmetrical Size and/or Reflects Octets modes being negotiated between Server and Control-Client in Unauthenticated mode, then because Sender DSCP and Sender ECN increase size of unauthenticated Session-Reflector packet by 4 octets the Padding Length value SHOULD be ≥ 28 octets to allow for the truncation process that TWAMP recommends in Section 4.2. 1 of [RFC5357].

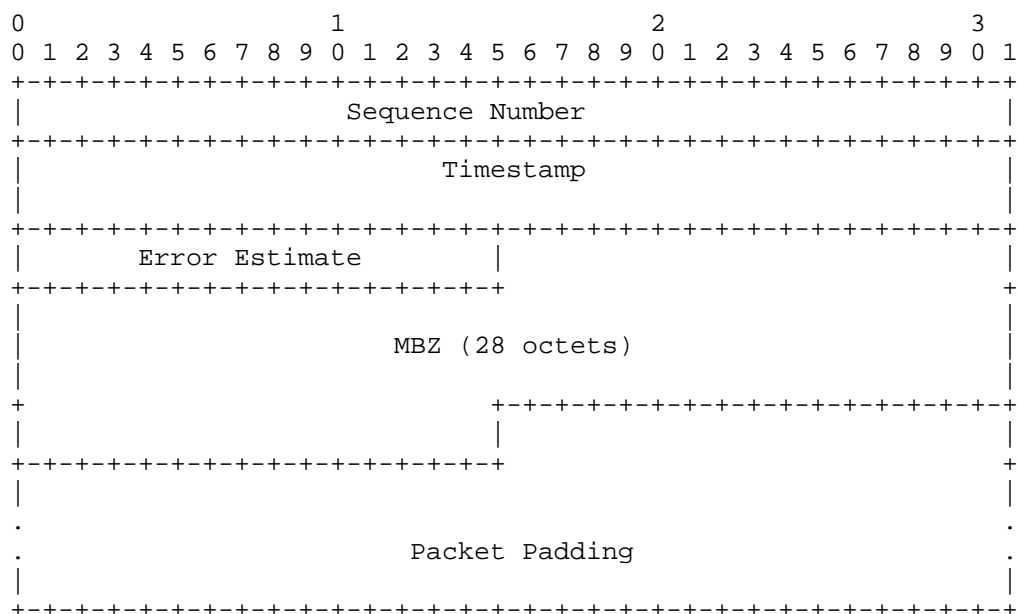


Figure 4: Session-Sender test packet format with DSCP and ECN monitoring and Symmetrical Test Packet in unauthenticated mode

2.2.3. Consideration for TWAMP Light mode

Appendix I of [RFC5357] does not explicitly state how value of Type-P descriptor synchronized between Session-Sender and Session-Reflector and whether different values considered as error condition and should be reported. We assume that by some means Session-Sender and Session-Reflector of given TWAMP-Test session informed to use the same DSCP value. Same means, i.e. configuration, could be used to inform Session-Reflector to support DSCP and ECN monitoring mode by copying data from received TWAMP test packets. Then Session-Sender may be informed to use Sender DSCP and ECN field in reflected TWAMP test packet.

3. IANA Considerations

The TWAMP-Modes registry defined in [RFC5618].

IANA is requested to reserve a new DSCP and ECN Monitoring Capability as follows:

Value	Description	Semantics	Reference
X (proposed 128)	DSCP and ECN Monitoring Capability	bit position Y (proposed 7)	This document

Table 1: New Type-P Descriptor Monitoring Capability

4. Security Considerations

Monitoring of DSCP and ECN does not appear to introduce any additional security threat to hosts that communicate with TWAMP as defined in [RFC5357], and existing extensions [RFC6038]. The security considerations that apply to any active measurement of live networks are relevant here as well. See the Security Considerations sections in [RFC4656] and [RFC5357].

5. Acknowledgements

Authors greatly appreciate thorough review and thoughtful comments by Chritofer Flinta and Samita Chakrabarti.

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.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [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.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the

Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, August 2009.

[RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, October 2010.

6.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Jonas Hedin
Ericsson

Email: jonas.hedin@ericsson.com

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Steve Baillargeon
Ericsson

Email: steve.baillargeon@ericsson.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 18, 2014

J. Fabini
Vienna University of Technology
A. Morton
AT&T Labs
October 15, 2013

Advanced Stream and Sampling Framework for IPPM
draft-ietf-ippm-2330-update-01

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 proposes to update the IP Performance Metrics (IPPM) Framework 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 April 18, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Definition: Reactive Path Behavior	3
2. Scope	4
3. New or Revised Stream Parameters	4
3.1. Test Packet Type-P	6
3.1.1. Multiple Test Packet Lengths	6
3.1.2. Test Packet Payload Content Optimization	6
3.2. Packet History	7
3.3. Access Technology Change	7
3.4. Time-Slotted Randomness Cancellation	7
4. Quality of Metrics and Methodologies	9
4.1. Repeatability	9
4.2. Continuity	10
4.3. Actionable	10
4.4. Conservative	11
4.5. Spatial and Temporal Composition	12
4.6. Poisson Sampling	12
5. Conclusions	12
6. Security Considerations	12
7. IANA Considerations	13
8. Acknowledgements	13
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Authors' Addresses	14

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. Although the proposed extensions can support methodologies to fulfill the continuity requirement stated in section 6.2 of [RFC2330], there is no guarantee. 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].

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 and the arrival characteristics at the reactive host/link depend on both the sending characteristics and the transfer characteristics of intervening hosts and links, then the reaction will appear less deterministic owing to the noise in the pattern at the reactive host/link.

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 scope of this memo is to describe useful stream parameters in addition to the information in Section 11.1 of [RFC2330] and described in [RFC3432] for periodic streams. The purpose 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.

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. This breaks the convention of payload correspondence when correlating measurements 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 effect 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.

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 either generated by a random device or by using part of a compressed file (e.g., a part of a ZIP compressed archive).

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.

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. 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

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. 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. 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. 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.

4.5. Spatial and Temporal Composition

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 measurement of subpaths is not feasible, restoring randomness of measurement samples when necessary is recommended as presented in [TSRC].

4.6. Poisson Sampling

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 to avoid reactive element state changes. 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 streams with alternate characteristics is generally discouraged with reactive paths because of the huge set of global parameters which have influence 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].

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

The authors thank Rudiger Geib, Matt Mathis and Konstantinos Pentikousis for their helpful comments on this draft.

9. References

9.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [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.
- [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.
- [RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.
- [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

- [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.
- [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.
- [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
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
URI: <http://home.comcast.net/~acmacm/>

IPPM WG
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

K. Pentikousis, Ed.
EICT
Y. Cui
E. Zhang
Huawei Technologies
October 21, 2013

Network Performance Measurement for IPsec
draft-ietf-ippm-ipsec-01

Abstract

IPsec is a mature technology with several interoperable implementations. Indeed, the use of IPsec tunnels is increasingly gaining popularity in several deployment scenarios, not the least in what used to be solely areas of traditional telecommunication protocols. Wider IPsec deployment 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. Unfortunately, however, standard IP performance measurement security mechanisms cannot be readily used with IPsec. This document makes the case for employing IPsec to protect the One-way and Two-Way Active Measurement Protocols (O/TWAMP) and proposes a method which combines IKEv2 and O/TWAMP as defined in RFC 4656 and RFC 5357, respectively. This specification aims, on the one hand, to ensure that O/TWAMP can be secured with the best mechanisms we have at our disposal today while, on the other hand, it facilitates the applicability of O/TWAMP to networks that have already deployed IPsec.

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 24, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Motivation	4
3.1. O/TWAMP-Control Security	5
3.2. O/TWAMP-Test Security	6
3.3. O/TWAMP Security Root	7
3.4. O/TWAMP and IPsec	7
4. O/TWAMP for IPsec Networks	8
4.1. Shared Key Derivation	8
4.2. Server Greeting Message Update	9
4.3. Session Key Derivation	11
4.3.1. Alternative 1	12
4.3.2. Alternative 2	14
5. Security Considerations	15
6. IANA Considerations	15
7. Acknowledgments	16
8. References	16
8.1. Normative References	16
8.2. Informative References	16
Authors' Addresses	16

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.

Cryptographic security mechanisms, such as IPsec, have been considered during the early stage of the specification of the two active measurement protocols mentioned above. However, due to several reasons, it was decided to avoid tying the development and deployment of O/TWAMP to such security mechanisms. In practice, for many networks, the issues listed in [RFC4656], Sec. 6.6 with respect to IPsec are still valid. However, we expect that in the near future IPsec will be deployed in many more hosts and networks than today. For example, IPsec tunnels may be used to secure wireless channels. In this case, what we are interested in is measuring network performance specifically for the traffic carried by the secured tunnel, not over the wireless channel in general. This document makes the case that O/TWAMP should be cognizant when IPsec and other security mechanisms are in place and can be leveraged upon. In other words, it is now time to specify how O/TWAMP is used in a network environment where IPsec is already deployed. We expect that in such an environment, measuring IP performance over IPsec tunnels with O/TWAMP is an important tool for operators.

For example, when considering the use of O/TWAMP in networks with IPsec deployed, we can take advantage of the IPsec key exchange protocol [RFC5996]. In particular, we note that it is not necessary to use distinct keys in OWAMP-Control and OWAMP-Test layers. One key for encryption and another for authentication is sufficient for both Control and Test layers. This obviates the need to generate two keys for each layer and reduces the complexity of O/TWAMP protocols in an IPsec environment. This observation comes from the fact that separate session keys in the OWAMP-Control and OWAMP-Test layers were designed for preventing reflection attacks when employing the current mechanism. Once IPsec is employed, such a potential threat is alleviated.

The remainder of this document is organized as follows. Section 3 motivates this work by revisiting the arguments made in [RFC4656] against the use of IPsec; this section also summarizes protocol operation with respect to security. Section 4 presents a method of binding O/TWAMP and IKEv2 for network measurements between a sender and a receiver which both support IPsec. Finally, Section 5 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. Motivation

In order to motivate the solutions proposed in this document, let us first revisit Section 6.6 of [RFC4656]. As we explain below, the reasons originally listed therein may not apply in many cases today.

RFC 4656 opts against using IPsec and instead favors the use of "a simple cryptographic protocol (based on a block cipher in CBC mode)". The first argument justifying this decision in [RFC4656] is that partial authentication in OWAMP authentication mode is not possible with IPsec. IPsec indeed cannot authenticate only a part of a packet. However, in an environment where IPsec is already deployed and actively used, partial authentication for OWAMP contradicts the operational reasons dictating the use of IPsec. It also increases the operational complexity of OWAMP (and TWAMP) in networks where IPsec is actively used and may in practice limit its applicability.

The second argument made is the need to keep separate deployment paths between OWAMP and IPsec. In several currently deployed types of networks IPsec is widely used to protect the data and signaling planes. For example, in mobile telecommunication networks, the deployment rate of IPsec exceeds 95% with respect to the LTE serving network. In older technology cellular networks, such as UMTS and GSM, IPsec use penetration is lower, but still quite significant. Additionally, there is a great number of IPsec-based VPN applications which are widely used in business applications to provide end-to-end security over, for instance, publicly open or otherwise untrusted IEEE 802.11 wireless LANs. At the same time, many IETF-standardized protocols make use of IPsec/IKE, including MIPv4/v6, HIP, SCTP, BGP, NAT and SIP, just to name a few.

The third argument in [RFC4656] is that, effectively, the adoption of IPsec in OWAMP may be problematic for "lightweight embedded devices." However, since the publication of RFC 4656, a large number of limited-resource and low-cost hardware, such as Ethernet switches, DSL modems, set-top boxes and other such devices come with support for IPsec "out of the box". Therefore concerns about implementation, although likely valid a decade ago, are not well founded today.

Finally, everyday use of IPsec applications by field technicians and good understanding of the IPsec API by many programmers should no longer be a reason for concern. On the contrary: By now, IPsec open source code is available for anyone who wants to use it. Therefore, although IPsec does need a certain level of expertise to deal with

it, in practice, most competent technical personnel and programmers have no problems using it on a daily basis.

OWAMP and TWAMP actually consist of two inter-related protocols: O/TWAMP-Control and O/TWAMP-Test. With respect to TWAMP, since "TWAMP and OWAMP use the same protocol for establishment of Control and Test procedures" [RFC5357] (Section 6), IPsec is also not considered. O/TWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results, whereas O/TWAMP-Test is used to exchange test packets between two measurement nodes.

In the remainder of this section we review security for O/TWAMP-Control and O/TWAMP-Test separately and then make the case for using them over IPsec.

3.1. O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on

- o AES in Cipher Block Chaining (AES-CBC) for confidentiality
- o HMAC-SHA1 truncated to 128 bits for message authentication

Three modes of operation are supported: unauthenticated, authenticated, and encrypted. The authenticated and encrypted 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 and encrypted modes, security parameters are negotiated during the control connection establishment.

Figure 1 illustrates the initiation stage of the O/TWAMP-Control protocol between a client and the server. In short, the client opens a TCP connection to the server in order to be able to send OWAMP-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 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 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.

```

+-----+           +-----+
| Client |           | Server |

```

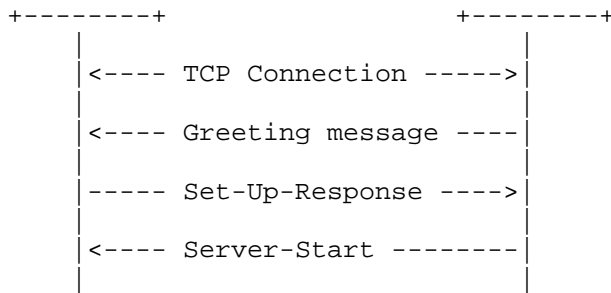


Figure 1: Initiation of O/TWAMP-Control

Encryption uses a key derived from the shared secret associated with KeyID. In the authenticated and encrypted modes, all further communication is encrypted using the AES Session-key and authenticated with the HMAC Session-key. The 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 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.

3.2. O/TWAMP-Test Security

The O/TWAMP-Test protocol runs over UDP, using the sender and receiver IP and port numbers that were negotiated during the Request-Session exchange. O/TWAMP-Test has the same three modes as with O/TWAMP-Control (unauthenticated, authenticated, and encrypted) and all O/TWAMP-Test sessions inherit the corresponding O/TWAMP-Control session 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 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 of 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.

3.3. O/TWAMP Security Root

As discussed above, the AES Session-key and HMAC Session-key used in the O/TWAMP-Test protocol are derived from the AES Session-key and HMAC Session-key which are used in O/TWAMP-Control protocol. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the 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/IPsec can automatically manage the security root and solve this problem, as we explain in Section 4.

3.4. O/TWAMP and IPsec

According to [RFC4656] the "deployment paths of IPsec and OWAMP could be separate if OWAMP does not depend on IPsec." However, the problem that arises in practice is that the security mechanism of O/TWAMP and IPsec cannot coexist at the same time without adding overhead or increasing complexity.

IPsec provides confidentiality and data integrity to IP datagrams. Distinct protocols are provided: Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE v1/v2). AH provides only integrity protection, while ESP can also provide encryption. IKE is used for dynamical key negotiation and automatic key management.

When sender and receiver implement O/TWAMP over IPsec, they need to agree on a shared secret key during the IPsec tunnel establishment. Subsequently, all IP packets sent by the sender are protected. If the AH protocol is used, IP packets are transmitted in plaintext.

The authentication part covers the entire packet. So all test information, such as UDP port number, and the test results will be visible to any attacker, which can intercept these test packets, and introduce errors or forge packets that may be injected during the transmission. In order to avoid this attack, the receiver must validate the integrity of these packets with the negotiated secret key. If ESP is used, IP packets are encrypted, and hence only the receiver can use the IPsec secret key to decrypt the IP packet, and obtain the test data in order to assess the IP network performance based on the measurements. Both the sender and receiver must support IPsec to generate the security secret key of IPsec.

Currently, after the test packets are received by the receiver, it cannot execute active measurement over IPsec. That is because the receiver knows only the shared secret key but not the IPsec key, while the test packets are protected by the IPsec key ultimately. Therefore, it needs to be considered how to measure IP network performance in an IPsec tunnel with O/TWAMP. Without this functionality, the use of OWAMP and TWAMP over IPsec is hindered.

Of course, backward compatibility should be considered as well. That is, the intrinsic security method based on shared key as specified in the O/TWAMP standards can also still be suitable for other network settings. There should be no impact on the current security mechanisms defined in O/TWAMP for other use cases. This document describes possible solutions to this problem which take advantage of the secret key derived by IPsec, in order to provision the key needed for active network measurements based on [RFC4656] and [RFC5357].

4. 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 [RFC5996].

4.1. Shared Key Derivation

If the AH protocol is used, the IP packets are transmitted in plaintext, but all O/TWAMP traffic is integrity-protected by IPsec. Therefore, even if the peers choose to opt for the unauthenticated mode, IPsec integrity protection is extended to O/TWAMP. In the authenticated and encrypted modes, the shared secret can be derived from the IKEv2 Security Association (SA), or IPsec SA.

If the shared secret key is derived from the IKEv2 SA, SKEYSEED must be generated firstly. SKEYSEED and its derivatives are computed as per [RFC5996], where prf is a pseudorandom function:

$$\text{SKEYSEED} = \text{prf}(\text{Ni} \mid \text{Nr}, \text{g}^{\text{ir}})$$

Ni and Nr are, respectively, the initiator and responder nonces, which are negotiated during the initial exchange (see Section 1.2 of [RFC5996]). g^{ir} is the shared secret from the ephemeral Diffie-Hellman exchange and is represented as a string of octets. Note that this SKEYSEED can be used as the O/TWAMP shared secret key directly.

Alternatively, the shared secret key can be generated as follows:

$$\text{Shared secret key} = \text{PRF}\{ \text{SKEYSEED}, \text{Session ID} \}$$

wherein the Session ID is the O/TWAMP-Test SID.

If the shared secret key is derived from the IPsec SA, instead, the shared secret key can be equal to KEYMAT, wherein

$$\text{KEYMAT} = \text{prf}+(\text{SK}_d, \text{Ni} \mid \text{Nr})$$

The term "prf+" stands for a function that outputs a pseudorandom stream based on the inputs to a prf, while SK_d is defined in [RFC5996] (Sections 2.13 and 1.2, respectively). The shared secret key can alternatively be generated as follows:

$$\text{Shared secret key} = \text{PRF}\{ \text{KEYMAT}, \text{Session ID} \}$$

wherein the session ID is the O/TWAMP-Test SID.

If rekeying for the IKE SA and IPsec SA occurs, the corresponding key of the SA is updated. Generally, ESP and AH SAs always exist in pairs, with one SA in each direction. If the SA is deleted, the key generated from the IKE SA or IPsec SA should also be updated.

4.2. Server Greeting Message Update

As discussed above, a binding association between the key generated from IPsec and the O/TWAMP shared secret key needs to be considered. The Security Association (SA) can be identified by the Security Parameter Index (SPI) and protocol uniquely for a given sender and receiver pair. Therefore, these parameters should be agreed upon during the initiation stage of O/TWAMP-Control. At the stage that the sender and receiver negotiate the integrity key, the IPsec protocol and SPI MUST be checked. Only if the two parameters are matched with the IPsec information, MUST the O/TWAMP connection be established.

The Security Parameter Index (SPI) and protocol type (see [RFC4301] [RFC5996]) will need to be included in the Server Greeting of the O/

TWAMP-Control protocol depicted in Figure 1. After the client receives the greeting, it **MUST** close the connection if it receives a greeting with an erroneous SPI and protocol value (Figure 2). Otherwise, the client **SHOULD** generate the shared secret key as discussed in Section 4.1 and respond with the server-expected Set-Up-Response message.

The Modes field in Figure 2 will need to allow for support of key derivation as discussed in Section 4.1. As such, pending discussion in the IPPM WG, Modes value 8 **MUST** be supported by compatible implementations, indicating support for IPsec. Server implementations compatible with this document **MUST** set the first 28 bits of the Modes field to zero. A client compatible with this specification **MUST** ignore the first 28 bits of the Modes field. For backward compatibility, the server is obviously allowed to indicate support for the Modes defined in [RFC4656]

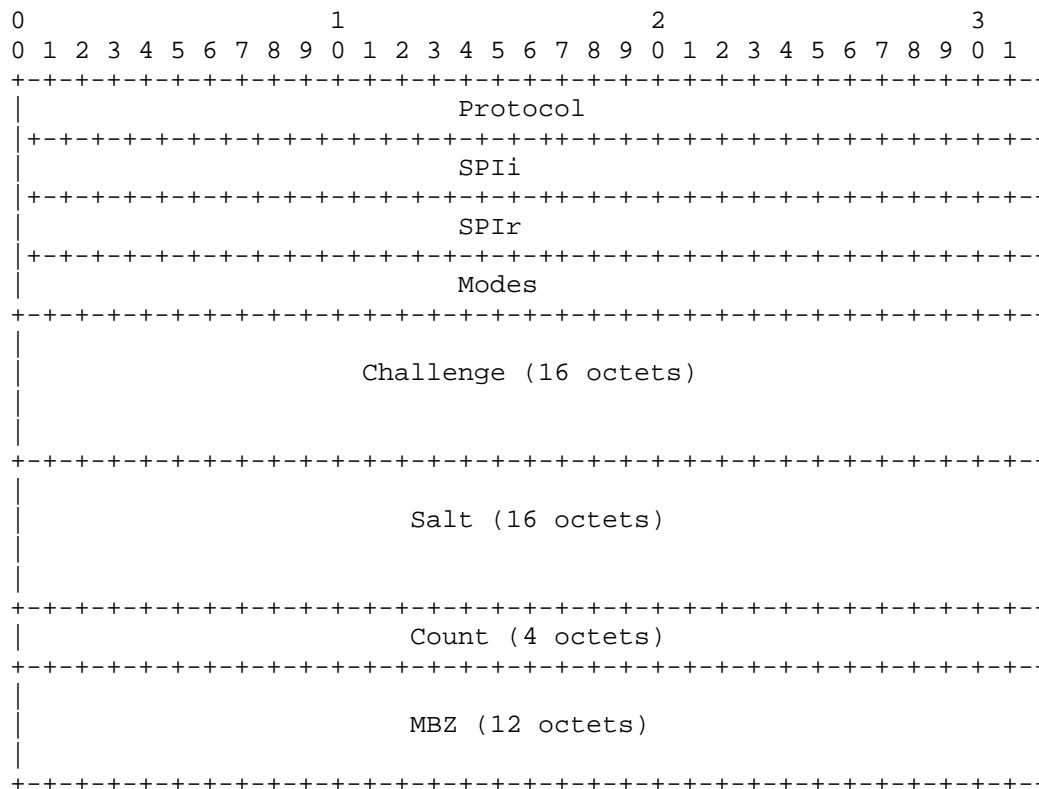


Figure 2: Server Greeting format

A compatible O/TWAMP client implementation would then interpret the originally unused 12 bits of the Server Greeting (see sec. 3.1 of [RFC4656]) as follows: The first 4 octets of the Server Greeting message indicate the protocol type, while the following 8 octets indicate the initiator (SPIi) and responder (SPIr) SPIs as illustrated in Figure 2. Note that in this case, the remaining fields of the Server Greeting message remain as per [RFC4656].

EDITOR'S NOTE:

We expect that this implementation option would pose the least backwards compatibility problems to existing O/TWAMP clients. Robust client implementations of [RFC4656] would disregard that the 29th Modes bit in the Server Greeting is set, and should ignore the information contained in the newly defined fields (Protocol, SPIi, SPIr). If the server supports other Modes, as one would assume, the client would then indicate any of the Modes defined in [RFC4656] and effectively indicate that it does not support the IPsec mode. At this point, the Server would need to use the Modes defined in [RFC4656] only.

When using ESP, all IP packets are encrypted, and therefore only the receiver can use the IPsec key to decrypt the IP active measurement packets. In this case, the IPsec tunnel between the sender and receiver provides additional security: even if the peers choose the unauthenticated mode, IPsec encryption and integrity protection is provided to O/TWAMP. If the sender and receiver decide to use the authenticated or encrypted mode, the shared secret can also be derived from IKE SA or IPsec SA. The method for key generation and binding association is the same discussed above for the AH protocol mode.

There is an encryption-only configuration in ESP, though this is not recommended due to its limitations. Since it does not produce integrity key in this case, either encryption-only ESP should be prohibited for O/TWAMP, or a decryption failure should be distinguished due to possible integrity attack.

4.3. Session Key Derivation

Section 4.1 described a method for deriving the shared key for O/TWAMP by capitalizing on IPsec. This is a step forward in terms of facilitating O/TWAMP deployment at scale in IPsec networks as it allows for greater and secure automation of standardized network performance measurements. We note, however, that the O/TWAMP protocol uses distinct encryption and integrity keys for O/TWAMP-Control and O/TWAMP-Test. Consequently, four keys are generated to protect O/TWAMP-Control and O/TWAMP-Test messages.

In fact, once IPsec is employed, one key for encryption and another for authentication is sufficient for both the Control and Test protocols. Therefore, in an IPsec environment we can further reduce the operational complexity of O/TWAMP protocols in a straightforward manner, as discussed below.

EDITOR'S NOTE:

We expect that both session key derivation proposals and optimization alternatives will be discussed in the IPPM working group and we are looking forward to community comments and feedback.

4.3.1. Alternative 1

If an IPsec SA is established between the server and the client, or both server and client support IPsec, the root key for O/TWAMP-based active network measurements can be derived from the IKE or IPsec SA.

If the root key that will be used in O/TWAMP network performance measurements is derived from the IKE SA, SKEYSEED must be generated first. SKEYSEED and its derivatives are computed as per [RFC5996]. SKEYSEED can be used as the root key of O/TWAMP directly; then the root key of O/TWAMP is equal to SKEYSEED. If the root key of O/TWAMP is derived from the IPsec SA, the shared secret key can be equal to KEYMAT. KEYMAT and its derivatives are computed as per usual [RFC5996].

Then, the session keys for encryption and authentication can be derived from the root key of O/TWAMP, wherein:

Session key for enc = PRF{ root key of O/TWAMP, "O/TWAMP enc" }

Session key for auth = PRF{ root key of O/TWAMP, "O/TWAMP auth" }

The former can provide encryption protection for O/TWAMP-Control and O/TWAMP-Test messages, while the latter can provide integrity protection.

Note that there are cases where rekeying the IKE SA and IPsec SA is necessary, and after which the corresponding key of SA is updated. If the SA is deleted, the O/TWAMP shared key generated from the IKE SA or IPsec SA should also be updated.

EDITOR'S NOTE:

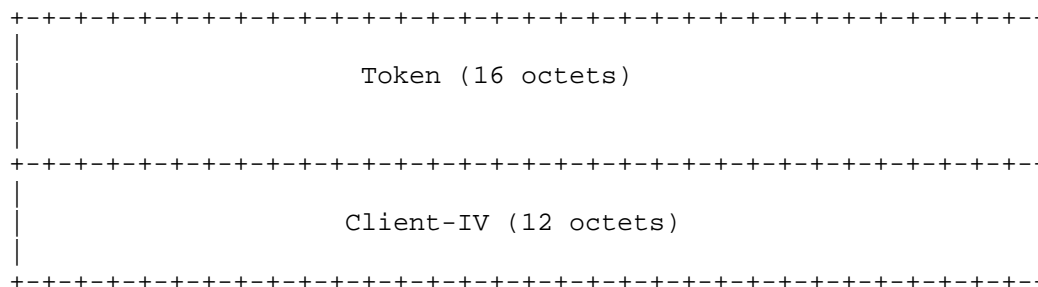


Figure 4: Set-Up-Response in Alternative 1

If the server authenticates the token successfully, then the O/TWAMP-Control message exchange flow can continue.

4.3.2. Alternative 2

Another way for optimizing the shared key use is to set the O/TWAMP session keys equal to the keys of the IPsec SA directly, i.e:

Session key for enc = encryption key of the IPsec SA

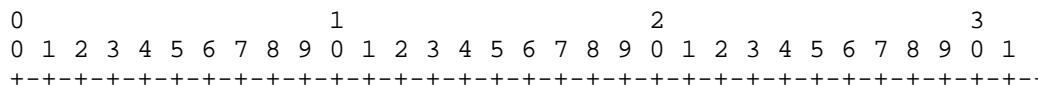
Session key for auth = integrity key of the IPsec SA

The former session key can provide encryption protection for O/TWAMP-Control and O/TWAMP-Test messages, while the latter can provide integrity protection. The point made in the previous subsection about rekeying the IPsec SA applies here too.

EDITOR'S NOTE:

As noted in the previous subsection, here too we can reduce the verbosity of the Server Greeting and Set-Up-Response messages even further, as explained below. Note, however, that such O/TWAMP message simplification poses backward compatibility challenges, which should be discussed in the IPPM WG.

The O/TWAMP control message exchange flow remains the same (i.e. as per Figure 1), while the Server Greeting format is illustrated in Figure 5. The Challenge, Salt, and Count parameters can be eliminated since the session keys of O/TWAMP are equal to the keys of an IPsec SA directly. SPI can identify the IPsec SA where the session keys derived from. The similarly optimized Set-Up-Response message is illustrated in Figure 6.



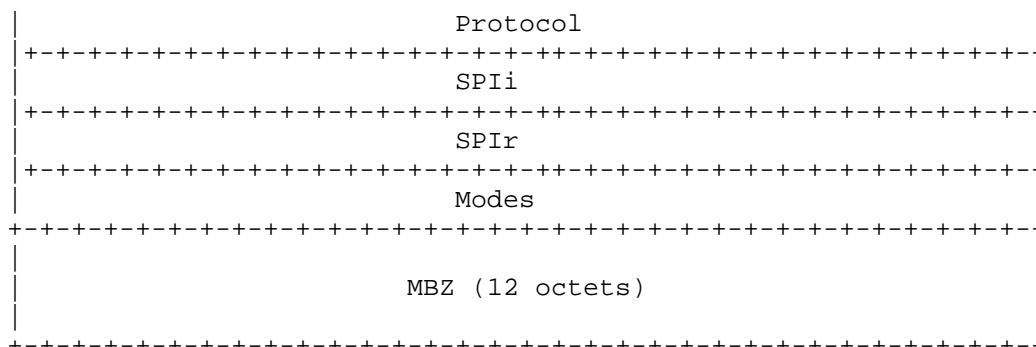


Figure 5: Optimized Server Greeting format

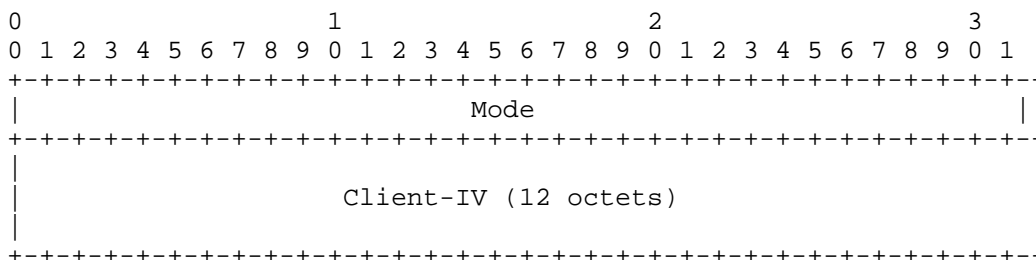


Figure 6: Set-Up-Response in Alternative 2

5. Security Considerations

As the shared secret key is derived from IPsec, the key derivation algorithm strength and limitations are as per [RFC5996]. 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 [RFC5996].

EDITOR'S NOTE:

As a general note, the IPPM community may want to revisit the arguments listed in [RFC4656], Sec. 6.6. Other widely-used Internet security mechanisms, such as TLS and DTLS, may also be considered for future use over and above of what is already specified in [RFC4656] [RFC5357].

6. IANA Considerations

IANA may need to allocate additional values for the Modes options presented in this document. The values of the protocol field may need to be assigned from the numbering space.

7. Acknowledgments

Emily Bi contributed to an earlier version of this document.

We thank Eric Chen and Yakov Stein for their comments on this draft, and Al Morton for the discussion on related earlier work in IPPM WG.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

8.2. Informative References

- [RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

Kostas Pentikousis (editor)
EICT GmbH
Torgauer Strasse 12-15
10829 Berlin
Germany

Email: k.pentikousis@eict.de

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

Email: cuiyang@huawei.com

Emma Zhang
Huawei Technologies
Huawei Building, Q20, No.156, Rd. BeiQing
Haidian District , Beijing 100095
P. R. China

Email: emma.zhanglijia@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2014

M. Bagnulo
UC3M
T. Burbridge
BT
S. Crawford
SamKnows
P. Eardley
BT
A. Morton
AT&T Labs
September 24, 2013

A Reference Path and Measurement Points for LMAP
draft-ietf-ippm-lmap-path-01

Abstract

This document defines a reference path for Large-scale Measurement of Broadband Access Performance (LMAP) and measurement points for commonly used performance 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 March 28, 2014.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Purpose and Scope	3
3. Terms and Definitions	3
3.1. Reference Path	3
3.2. Subscriber	3
3.3. Dedicated Component (Links or Nodes)	4
3.4. Shared Component (Links or Nodes)	4
3.5. Resource Transition Point	4
4. Reference Path	4
5. Measurement Points	6
6. Translation Between Ref. Path and Tech. X	7
7. Example Resource Transition	9
8. Security considerations	10
9. IANA Considerations	10
10. Acknowledgements	10
11. References	10
11.1. Normative References	10
11.2. Informative References	11
Authors' Addresses	11

1. Introduction

This document defines a reference path for Large-scale Measurement of Broadband Access Performance (LMAP). 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 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] (which may also figure in the LMAP work effort). Section 4.1 of [RFC5835] defines the term "measurement point".

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

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 metadata to describe measurement results. Measurements conducted over different path scopes are not a valid basis for performance comparisons.

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 without ambiguity.

The bridge between the reference path and specific network technologies (with differing underlying architectures) is within the scope of this effort. 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 this regard. This should serve many measurement uses, including diagnostic (where the same metric may be measured over many different path scopes) and comparative (where the same metric may be measured on different network infrastructures).

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 routers, switches, links, radios, and processing elements that comprise all the network elements traversed by each packet between the source and destination hosts. The reference path is intended to be equally applicable to all networking technologies, therefore the components are generically defined, but their functions should have a clear counterpart or be obviously omitted in any network technology.

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 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.

4. Reference Path

This section defines a reference path for Internet Access.

```
Subsc. -- Private -- Private -- Access -- 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 Access Service Subscriber. In some configurations, one or more private networks and the device that provides the Access 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 Access (Service) Demarcation point - this varies by technology but is usually defined as the Ethernet interface on a residential gateway or modem where the scope of access 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., LTE UE). Ownership may not 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 access network, or a third-party will certify standards compliance.
- o Intra IP Access - This is the first point in the access architecture beyond the Access 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 GRA GW. This point could also collapse to the device providing the Access Service Demarc., in principle. Only one Intra IP Access point is shown, but they can be identified in any access or transit network.
- o GRA GW - the point of interconnection between the access administrative domain and the rest of the Internet, where routing will depend on the GRAs in the IP header.
- o Transit GRA GW - Networks that intervene between the Subscriber's Access network and the Destination Host's network are designated "transit" and involve two 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 Access Service Demarc. and the Intra IP Access points must use the same address space and be separated by the shared

and dedicated access link infrastructure, such that a test between these points produces a useful assessment of access performance.

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, deterministic errors that can be quantified are ideal.

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

Subsc.	--	Private	--	Private	--	Access	--	Intra IP	--	GRA	--	Transit
device		Net #1		Net #2		Demarc.		Access		GW		GRA GW
mp000						mp100		mp150		mp190		mp200
...												
Transit	--	GRA	--	Service	--	Private	--	Private	--	Destination		
GRA GW		GW		Demarc.		Net #n		Net #n+1		Host		
mpX90		mp890		mp800						mp900		

GRA = Globally Routable Address, GW = Gateway

The numbering for measurement points (mpNNN) allows for considerable local use of unallocated numbers.

Notes:

- o Some use the terminology "on-net" and "off-net" when referring to Internet Service Provider (ISP) measurement coverage. With respect to the reference path, tests between mp100 and mp190 are "on-net".
- o Widely deployed broadband 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 used at all measurement points must be indicated, especially the interface standard and configured speed.
- o If it can be shown that a link connecting to a measurement point has reliably deterministic or negligible performance, then the remote end of the connecting link is an equivalent point for some methods of measurement (To Be Specified Elsewhere). In any case, the presence of such a link must be reported.
- o Many access network architectures have a traffic aggregation point (e.g., CMTS or DSLAM) between mp100 and mp150. We designate this point mp120, but it won't currently fit in the figure.
- o A Carrier Grade NAT (CGN) deployed in the Subscriber's access network would be positioned between mp100 and mp190, and the egress side of the CGN will typically be designated mp150.
- o In the case that a private address space is used in an access architecture, then mp100 may need to use the same address space as its remote measurement point counterpart, so that a test between these points produces a useful assessment of network performance. Tests between mp000 and mp100 could use private address space, and when the egress 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.

6. Translation Between Ref. Path and Tech. X

This section and those that follow are intended to provide a more exact mapping between particular network technologies and the reference path.

We provide an example for 3G Cellular access below.

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

	_____UE_____		____RAN+Core____		____GGSN____	
--	--------------	--	------------------	--	--------------	--

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

We next provide a few examples of DSL access. Consider first the case where:

- o The Customer Premises Equipment (CPE) is 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 though 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	--	Access	--	Intra IP	--	GRA	--	Transit
device		Net #1		Net #2		Demarc.		Access		GW		GRA GW
mp000						mp100		mp150		mp190		mp200

	--UE--		-----CPE/NAT-----		-----		BRAS-		-----	
							----Access Network--			

GRA = Globally Routable Address, GW = Gateway

Consider next the 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 into 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 though the WiFi access.

We believe 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 mp000	-- Private Net #1	-- Private Net #2	-- Access Demarc. mp100	-- Intra IP Access mp150	-- GRA GW mp190	-- Transit GRA GW mp200
--UE--	-----CPE/NAT-----		-----	-CGN-	-----	
				---Access Network---		

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 is 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 though the WiFi access.
- o The Wi-Fi subnetwork (Private Net#1) shares unlicensed radio channel resources with other W-Fi 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 Access 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 is becoming a fairly common configuration in parts of the world.

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

Subsc. device	-- Private Net #1	-- Private Net #2	-- Access Demarc.	-- Intra IP Access	-- GRA GW	-- Transit GRA GW
mp000			mp100	mp150	mp190	mp200
--UE--	-----CPE/NAT-----		-----	-CGN-	-----	
	Wi-Fi	wired		---Access Network---		
	-Shared-- RT		-----Dedicated-----	RT	-----Shared-----...	

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 readers of this memo or to the Internet itself.

9. IANA Considerations

TBD

10. Acknowledgements

Thanks to Matt Mathis 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.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, August 2012.

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [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.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.

11.2. Informative References

- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.
- [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.

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>

Trevor Burbridge
British Telecom
Adastral Park, Martlesham Heath
IPswitch
ENGLAND

Email: trevor.burbridge@bt.com

Sam Crawford
SamKnows

Email: sam@samknows.com

Phil Eardley
British Telecom
Adastral Park, Martlesham Heath
IPswitch
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: April 24, 2014

M. Mathis
Google, Inc
A. Morton
AT&T Labs
October 21, 2013

Model Based Bulk Performance Metrics
draft-ietf-ippm-model-based-metrics-01.txt

Abstract

We introduce a new class of model based metrics designed to determine if a long network path can meet predefined end-to-end application performance targets by applying a suite of IP diagnostic tests to successive subpaths. The subpath at a time tests are designed to exclude all known conditions which might prevent the full end-to-end path from meeting the user's target application performance.

This approach makes it possible to to determine the IP performance requirements needed to support the desired end-to-end TCP performance. The IP metrics are based on traffic patterns that mimic TCP or other transport protocol but are precomputed independently of the actual behavior of the transport protocol over the subpath under test. This makes the measurements open loop, eliminating nearly all of the difficulties encountered by traditional bulk transport metrics, which fundamentally depend on congestion control equilibrium behavior.

A natural consequence of this methodology is verifiable network measurement: measurements from any given vantage point can be verified by repeating them from other vantage points.

Formatted: Mon Oct 21 15:42:35 PDT 2013

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 24, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	5
1.1. TODO	6
2. Terminology	6
3. New requirements relative to RFC 2330	9
4. Background	10
4.1. TCP properties	12
5. Common Models and Parameters	14
5.1. Target End-to-end parameters	14
5.2. Common Model Calculations	15
5.3. Parameter Derating	16
6. Common testing procedures	16
6.1. Traffic generating techniques	16
6.1.1. Paced transmission	16
6.1.2. Constant window pseudo CBR	17
6.1.3. Scanned window pseudo CBR	18
6.1.4. Concurrent or channelized testing	18
6.1.5. Intermittent Testing	19
6.1.6. Intermittent Scatter Testing	20
6.2. Interpreting the Results	20
6.2.1. Test outcomes	20
6.2.2. Statistical criteria for measuring run_length	21
6.2.3. Reordering Tolerance	23
6.3. Test Qualifications	23
6.3.1. Verify the Traffic Generation Accuracy	23
6.3.2. Verify the absence of cross traffic	24
6.3.3. Additional test preconditions	25
7. Diagnostic Tests	25
7.1. Basic Data Rate and Run Length Tests	25
7.1.1. Run Length at Paced Full Data Rate	26
7.1.2. run length at Full Data Windowed Rate	26
7.1.3. Background Run Length Tests	26
7.2. Standing Queue tests	26
7.2.1. Congestion Avoidance	28
7.2.2. Bufferbloat	28
7.2.3. Non excessive loss	28
7.2.4. Duplex Self Interference	28
7.3. Slowstart tests	29
7.3.1. Full Window slowstart test	29
7.3.2. Slowstart AQM test	29
7.4. Sender Rate Burst tests	29
7.5. Combined Tests	30
7.5.1. Sustained burst test	30
7.5.2. Live Streaming Media	31
8. Examples	32
8.1. Near serving HD streaming video	32
8.2. Far serving SD streaming video	32

8.3. Bulk delivery of remote scientific data	33
9. Validation	33
10. Acknowledgements	34
11. Informative References	35
Appendix A. Model Derivations	36
A.1. Aggregate Reno	37
A.2. CUBIC	37
Appendix B. Version Control	38
Authors' Addresses	38

1. Introduction

Model based bulk performance metrics evaluate an Internet path's ability to carry bulk data. TCP models are used to design a targeted diagnostic suite (TDS) of IP performance tests which can be applied independently to each subpath of the full end-to-end path. A targeted diagnostic suite is constructed such that independent tests of the subpaths will accurately predict if the full end-to-end path can deliver bulk data at the specified performance target, independent of the measurement vantage points or other details of the test procedures used to measure each subpath.

Each test in the TDS consists of a precomputed traffic pattern and statistical criteria for evaluating packet delivery.

TCP models are used to design traffic patterns that mimic TCP or other bulk transport protocol operating at the target performance and RTT over a full range of conditions, including flows that are bursty at multiple time scales. The traffic patterns are computed in advance based on the properties of the full end-to-end path and independent of the properties of individual subpaths. As much as possible the traffic is generated deterministically in ways that minimizes the extent to which test methodology, measurement points, measurement vantage or path partitioning effect the details of the traffic.

Models are also used to compute the bounds on the packet delivery statistics for acceptable the IP performance. The criteria for passing each test are determined from the end-to-end target performance and are independent of the subpath under test. In addition to passing or failing, a test can be inconclusive if the precomputed traffic pattern was not authentically generated, test preconditions were not met or the measurement results were not statistically significant.

TCP's ability to compensate for less than ideal network conditions is fundamentally affected by the RTT and MTU of the end-to-end Internet path that it traverses. The end-to-end path determines fixed bounds on these parameters. The target values for these three parameters, Data Rate, RTT and MTU, are determined by the application, its intended use and the physical infrastructure over which it is intended to traverse. These parameters are used to inform the models used to design the TDS.

This document describes a framework for deriving the traffic and delivery statistics for model based metrics. It does not fully specify any measurement techniques. Important details such as packet type-p selection, sampling techniques, vantage selection, etc are out

of scope for this document. We imagine Fully Specified Targeted Diagnostic Suites (FSTDs), that fully defines all of these details. We use TDS to refer to the subset of such a specification that is in scope for this document. A TDS includes specification for the traffic and delivery statistics for the diagnostic tests themselves, documentation of the models and any assumptions or derating used to derive the test parameters and a description of the test setup used to calibrate the models, as described in later sections.

Section 2 defines terminology used throughout this document.

It has been difficult to develop BTC metrics due to some overlooked requirements described in Section 3 and some intrinsic problems with using protocols for measurement, described in Section 4.

In Section 5 we describe the models and common parameters used to derive the targeted diagnostic suite. In Section 6 we describe common testing procedures. Each subpath is evaluated using suite of far simpler and more predictable diagnostic tests described in Section 7. In Section 8 we present three example TDS, one that might be representative of HD video, when served fairly close to the user, a second that might be representative of standard video, served from a greater distance, and a third that might be representative of an network designed to support high performance bulk download.

There exists a small risk that model based metric itself might yield a false pass result, in the sense that every subpath of an end-to-end path passes every IP diagnostic test and yet a real application falls to attain the performance target over the end-to-end path. If this happens, then the validation procedure described in Section 9 needs to be used to prove and potentially revise the models.

Future document will define model based metrics for other traffic classes and application types, such as real time streaming media.

1.1. TODO

Please send comments on this draft to 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: Mon Oct 21 15:42:35 PDT 2013

2. Terminology

Terminology about paths, etc. See [RFC2330] and [I-D.morton-ippm-lmap-path].

[data] sender Host sending data and receiving ACKs, typically via TCP.

[data] receiver Host receiving data and sending ACKs, typically via TCP.

subpath A portion of the full path. Note that there is no requirement that subpaths be non-overlapping.

Measurement Point Measurement points as described in [I-D.morton-ippm-lmap-path].

test path A path between two measurement points that includes a subpath of the end-to-end path under test, plus possibly additional infrastructure between the measurement points and the subpath.

[Dominant] Bottleneck The Bottleneck that determines a flow's self clock. It generally determines the traffic statistics for the entire path. See Section 4.1.

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 resources (network and/or queue capacity).

Properties determined by the end-to-end path and application. They 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. This is the payload data rate, and excludes TCP/IP (or other protocol) headers and retransmits.

Link Data Rate General term for the data rate as seen by the link or lower layers. It includes transport and IP headers, retransmits and other transport layer overhead. This document is agnostic as to whether the link data rate includes or excludes framing, MAC or other lower layer overheads, except that they must be treated uniformly.

end-to-end target parameters: Application or transport performance goals for the end-to-end path. They include the target data rate, RTT and MTU described below.

Target Data Rate: The application or ultimate user's performance goal. When converted to link data rate, it must be slightly smaller than the actual link data rate, otherwise there is no margin for compensating for RTT or other path properties. These test will be excessively brittle if the target data rate does not include any built in headroom.

Target RTT (Round Trip Time): The baseline (minimum) RTT of the longest end-to-end path the over which the application expects to meet the target performance. This must be specified considering authentic packets sizes: MTU sized packets on the forward path, header_overhead sized packets on the return (ACK) path.

Target MTU (Maximum Transmission Unit): The maximum MTU supported by the end-to-end path the over which the application expects to meet the target performance. Assume 1500 Bytes per packet unless otherwise specified. If some subpath forces a smaller MTU, then it becomes the target MTU, and all model calculations and subpath tests must use the same smaller MTU.

Effective Bottleneck Data Rate: This is the bottleneck data rate that might be inferred from the ACK stream, by looking at how much data the ACK stream reports was delivered per unit time. See Section 4.1 for more details.

[sender] [interface] rate: The burst data rate, constrained by the data sender's interfaces. 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 acknowledgements (ACKs). For TCP, the Maximum Segment Size (MSS) is the Target MTU minus the header overhead.

Basic parameters common to models and subpath tests. They are described in more detail in Section 5.2.

pipe size A general term for number of packets needed in flight (the window size) to exactly fill some network path or subpath. This is the window size which in normally the onset of queueing.

target_pipe_size: The number of packets in flight (the window size) needed to exactly meet the target rate, with a single stream and no cross traffic for the specified target data rate, RTT and MTU.

run length A general term for the observed, measured or specified number of packets that are (to be) delivered between losses or ECN marks. Nominally one over the loss or ECN marking probability.

target_run_length Required run length computed from the target data rate, RTT and MTU.

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 TDS validation procedures, described in Section 9.

`subpath_data_rate` The maximum IP data rate supported by a subpath. This typically includes TCP/IP overhead, including headers, retransmits, etc.

`test_path_RTT` The RTT (using appropriate packet sizes) between two measurement points.

`test_path_pipe` The amount of data necessary to fill a test path. Nominally the test path RTT times the `subpath_data_rate` (which should be part of the end-to-end subpath).

`test_window` The window necessary to meet the `target_rate` over a subpath. Typically `test_window=target_data_rate*test_RTT/target_MTU`.

Tests can be classified into groups according to their applicability

Capacity tests determine if a network subpath has sufficient capacity to deliver the target performance. As long as the test traffic is within the proper envelope for the target end-to-end performance, the average packet losses or ECN must be below the threshold computed by the model. As such, they reflect parameters that can transition from passing to failing as a consequence of additional presented load or the actions of other network users. By definition, capacity tests also consume significant network resources (data capacity and/or buffer space), and the test schedules must be balanced by their cost.

Monitoring tests are design to capture the most important aspects of a capacity test, but without causing unreasonable ongoing load themselves. As such they may miss some details of the network performance, but can serve as a useful reduced cost proxy for a capacity test.

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 marks. These tests are likely to have complicated interactions with other traffic and under some conditions can be inversely sensitive to load. For example a test to verify that an AQM algorithm causes ECN marks or packet drops early enough to limit queue occupancy may experience a false pass results in the presence of bursty 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.

3. New requirements relative to RFC 2330

[Move this entire section to a future paper]

Model Based Metrics are designed to fulfill some additional requirement that were not recognized at the time RFC 2330 [RFC2330] was written. These missing requirements may have significantly contributed to policy difficulties in the IP measurement space. Some additional requirements are:

- o 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 must be vantage point invariant over a significant range of measurement point choices (e.g., measurement points as described in [I-D.morton-ippm-lmap-path]), including off path measurement points. The only requirements on MP selection should be that the portion of the path that is not under test is effectively ideal (or is non ideal in calibratable ways) and the RTT between MPs is below some reasonable bound.
- o Metrics must be repeatable by multiple parties. It must 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.

NB: All of the metric requirements in RFC 2330 should be reviewed and potentially revised. If such a document is opened soon enough, this entire section should be dropped.

4. Background

[Move to a future paper, abridge here,]

At the time the IPPM WG was chartered, sound Bulk Transport Capacity measurement was known to be beyond our capabilities. By hindsight it is now clear why it 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.
- o Congestion control is an equilibrium process, transport protocols change the network (raise loss probability and/or RTT) to conform to their behavior.
- o TCP's ability to compensate for network flaws is directly proportional to the number of roundtrips per second (i.e. inversely proportional to the RTT). As a consequence a flawed link may pass a short RTT local test even though it fails when the path is extended by a perfect network to some larger RTT.
- o TCP has a meta Heisenberg problem - 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 the relative momentum of the measurement and

measured particles. For network measurement you can not in general determine the relative "elasticity" of the measurement traffic and cross traffic, so you can not even gage the relative magnitude of their effects on each other.

The MBM approach is to "open loop" TCP by precomputing traffic patterns that are typically generated by TCP operating at the given target parameters, and evaluating delivery statistics (losses, ECN marks and delay). In this approach the measurement software explicitly controls the data rate, transmission pattern or cwnd (TCP's primary congestion control state variables) to create repeatable traffic patterns that mimic TCP behavior but are independent of the actual network behavior of the subpath under test. These patterns are manipulated to probe the network to verify that it can deliver all of the traffic patterns that a transport protocol is likely to generate under normal operation at the target rate and RTT.

Models are used to determine the actual test parameters (burst size, loss rate, etc) from the target parameters. The basic method is to use models to estimate specific network properties required to sustain a given transport flow (or set of flows), and using a suite of metrics to confirm that the network meets the required properties.

A network is expected to be able to sustain a Bulk TCP flow of a given data rate, MTU and RTT when the following conditions are met:

- o The raw link rate is higher than the target data rate.
- o The raw packet run length is larger than required by a suitable TCP performance model
- o There is sufficient buffering at the dominant bottleneck to absorb a slowstart rate burst large enough to get the flow out of slowstart at a suitable window size.
- o 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 other mechanisms.
- o When there is a standing queue at a bottleneck for a shared media subpath, there are suitable bounds on how the data and ACKs interact, for example due to the channel arbitration mechanism.
- o 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.

The tests to verify these condition are described in Section 7.

A singleton [RFC2330] measurement is a pass/fail evaluation of a given path or subpath at a given performance. Note that measurements to confirm that a link passes at one particular performance might not be useful to predict if the link will pass at a different

performance.

A TDS does have several valuable properties, such as natural ways to define several different composition metrics [RFC5835].

[Add text on algebra on metrics (A-Frame from [RFC2330]) and tomography.] The Spatial Composition of fundamental IPPM metrics has been studied and standardized. For example, the algebra to combine empirical assessments of loss ratio to estimate complete path performance is described in section 5.1.5. of [RFC6049]. We intend to use this and other composition metrics as necessary.

We are developing a tool that can perform many of the tests described here[MBMSource].

4.1. TCP properties

[Move this entire section to a future paper]

TCP and SCTP are self clocked protocols. The dominant steady state behavior is to have an approximately fixed quantity of data and acknowledgements (ACKs) circulating in the network. The receiver reports arriving data by returning ACKs to the data sender, the data sender most frequently responds by sending exactly the same quantity of data back into the network. The 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 widow by sending slightly more or less data in response to each ACK. The fundamentally important property of this systems is that it is entirely 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 phenomena can cause bursts of data, even in idealized networks that are modeled as simple queueing systems.

During slowstart the data rate is doubled on each RTT by sending twice as much data as was delivered to the receiver on the prior RTT. For slowstart to be able to fill such a network 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 mark. For example, with classic Reno congestion control, an optimal slowstart has to end with a burst that is twice the bottleneck rate for exactly one RTT in duration. This burst causes a queue which is exactly equal to the pipe size (the window is exactly twice the pipe size) so when the window is halved, the new window will be exactly the pipe size.

Another source of bursts are application pauses. If the application pauses (stops reading or writing data) for some fraction of one RTT, state-of-the-art TCP to "catches up" to the earlier window size by sending a burst of data at the full sender interface rate. To fill such a network with a realistic application, the network has to be able to tolerate interface rate bursts from the data sender large enough to cover application pauses.

Note that if the bottleneck data rate is significantly slower than the rest of the path, the slowstart bursts will not cause significant queues anywhere else along the path; they primarily exercise the queue at the dominant bottleneck. Furthermore, although the interface rate bursts caused by the application are likely to be smaller than last burst of a slowstart, they are at a higher rate so they can exercise queues at arbitrary points along the "front path" from the data sender up to and including the queue at the bottleneck.

For many network technologies a simple queueing model does not apply: the network schedules, thins or otherwise alters the timing of ACKs and data, generally to raise the efficiency of the channel allocation process when confronted with relatively widely spaced small ACKs. These efficiency strategies are ubiquitous for half duplex, wireless or broadcast media.

Altering the ACK stream generally has two consequences: raising the effective bottleneck data rate making slowstart burst at higher rates (possibly as high as the sender's interface rate) and effectively raising the RTT by the time that the ACKs were postponed. The first effect can be partially mitigated by reclocking 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 class of behaviors is a half duplex channel that is never released until the current end point has no pending traffic. Such environments cause self clocked protocols revert to extremely inefficient stop and wait behavior, where they send an entire window of data as a single burst, followed by the entire window of ACKs on the return path.

If a particular end-to-end path contains a link or device that alters the ACK stream, then the entire 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 Effective Bottleneck Data Rate, which is the average rate at which the ACKs advance `snd.una`. Note that thinning the ACKs (relying on the cumulative nature of `seg.ack` to permit discarding some ACKs) is implies an effectively infinite bottleneck data rate.

To verify that a path can meet the performance target, it is

necessary to independently confirm that the entire path can tolerate bursts in the dimensions that are likely to be induced by the application and any data or ACK scheduling anywhere in the path. Two common cases are the most important: slowstart bursts at twice the effective bottleneck data rate; and somewhat smaller sender interface rate bursts.

The slowstart rate bursts must be at least as large as `target_pipe_size` packets and should be twice as large (so the peak queue occupancy at the dominant bottleneck would be approximately `target_pipe_size`).

There is no general model for how well the network needs to tolerate sender interface rate bursts. All existing TCP implementations send full sized full rate bursts under some typically uncommon conditions, such as application pauses that approximately match the RTT, or when ACKs are lost or thinned. Strawman: partial window bursts (some fraction of `target_pipe_size`) should be tolerated without significantly raising the loss probability. Full `target_pipe_size` bursts may slightly increase the loss probability. Interface rate bursts as large as twice `target_pipe_size` should not cause deterministic packet drops.

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 2. These parameters are determined by the needs of the application or the ultimate end user and the end-to-end Internet path over which the application is expected to operate. The target parameters are in units that make sense to the upper layer: payload bytes delivered to the application, above TCP. They exclude overheads associated with TCP and IP headers, retransmits and other protocols (e.g. DNS). In addition, other end-to-end parameters include the effective bottleneck data rate, the sender interface data rate and the TCP/IP header sizes (overhead).

Note that the target parameters can be specified for a hypothetical path, for example to construct TDS designed for bench testing in the absence of a real application, or for a real physical test, for in situ testing of production infrastructure.

The number of concurrent connections is explicitly not a parameter to this model [unlike earlier drafts]. If a subpath requires multiple connections in order to meet the specified performance, that must be

stated explicitly and the procedure described in Section 6.1.4 applies.

5.2. Common Model Calculations

The most important derived parameter is `target_pipe_size` (in packets), which is the window size --- the number of packets needed exactly meet the target rate, with no cross traffic for the specified target RTT and MTU. It is given by:

$$\text{target_pipe_size} = \text{target_rate} * \text{target_RTT} / (\text{target_MTU} - \text{header_overhead})$$

If the transport protocol (e.g. TCP) average window size is smaller than this, it will not meet the target rate.

The reference `target_run_length`, is a very conservative model for the minimum required spacing between losses or ECN marks. The reference `target_run_length` can be derived as follows: assume the `subpath_data_rate` is infinitesimally larger than the `target_data_rate` plus the required header overheads. Then `target_pipe_size` also predicts the onset of queueing. If the transport protocol (e.g. TCP) has a window size that is larger than the `target_pipe_size`, the excess packets will raise the RTT, typically by forming a standing queue at the bottleneck.

Assume the transport protocol is using standard Reno style Additive Increase, Multiplicative Decrease congestion control [RFC5681] and the receiver is using standard delayed ACKs. With delayed ACKs there must be $2 * \text{target_pipe_size}$ roundtrips between losses. Otherwise the multiplicative window reduction triggered by a loss would cause the network to be underfilled. We derive the number of packets between losses from the area under the AIMD sawtooth following [MSM097]. They must be no more frequent than every 1 in $(3/2) * \text{target_pipe_size} * (2 * \text{target_pipe_size})$ packets. This simplifies to:

$$\text{target_run_length} = 3 * (\text{target_pipe_size}^2)$$

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 less conservative model is used, a fully specified TDS or FSTDS MUST document the actual method for computing `target_run_length` along with the rationale for the underlying assumptions and the ratio of chosen `target_run_length` to the reference `target_run_length` calculated above.

These two parameters, `target_pipe_size` and `target_run_length`, directly imply most of the individual parameters for the tests below. `Target_pipe_size` is the window size, the amount of circulating data required to meet the target data rate, and implies the scale of the bursts that the network might experience. `Target_run_length` is the amount of data required between losses or ECN marks standard for standard congestion control.

The individual parameters for each diagnostic test is described below. In a few cases there are not well established models for what is considered correct network operation. In many of these cases the problems might either be partially mitigated by future improvements to TCP implementations.

5.3. Parameter Derating

Since some aspects of the models are very conservative, this 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 TDS or FSTDs MUST document and justify the actual method used compute the derated metric parameters.
- o The validation procedures described in Section 9 must be used to demonstrate the feasibility of meeting the performance targets with infrastructure that infinitesimally passes the derated tests.
- o The validation process 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 include derating as a way to indicate flexibility in the parameters.

6. Common testing procedures

6.1. Traffic generating techniques

6.1.1. Paced transmission

Paced (burst) transmissions: send bursts of data on a timer to meet a particular target rate and pattern. In all cases the specified data rate can either be the application or link rates. Header overheads must be included in the calculations as appropriate.

Paced single packets: Send individual packets at the specified rate or headway.

Burst: Send sender interface rate bursts on a timer. Specify any 3 of: average rate, packet size, burst size (number of packets) and burst headway (burst start to start). These bursts are typically sent as back-to-back packets at the testers interface rate.

Slowstart bursts: Send 4 packet sender interface rate bursts at an average data rate equal to twice effective bottleneck link rate (but not more than the sender interface rate). This corresponds to the average rate during a TCP slowstart when Appropriate Byte Counting [ABC] is present or delayed ack is disabled.

Repeated Slowstart bursts: Slowstart bursts are typically part of larger scale pattern of repeated bursts, such as sending `target_pipe_size` packets as slowstart bursts on a `target_RTT` headway (burst start to burst start). Such a stream has three different average rates, depending on the averaging time scale. At the finest time scale the average rate is the same as the sender interface rate, at a medium scale the average rate is twice the effective bottleneck link rate and at the longest time scales the average rate is the target data rate.

Note that if the effective bottleneck link rate is more than half of the sender interface rate, slowstart bursts become sender interface rate bursts.

6.1.2. Constant window pseudo CBR

Implement pseudo constant bit rate by running a standard protocol such as TCP with a fixed bound on the window size. The rate is only maintained in average over each RTT, and is subject to limitations of the transport protocol.

The bound on the window size is computed from the `target_data_rate` and the actual RTT of the test path.

If the transport protocol fails to maintain the test rate within prescribed data rates, the test MUST NOT be considered passing. If there is a signature of a network problem (e.g. the run length is too small) then the test can be considered to fail. Since packet loss and ECN marks are required to reduce the data rate for standard transport protocols, the test specification must include suitable allowances in the prescribed data rates. If there is not sufficient signature of a network problem, then failing to make the prescribed data rate must be considered inconclusive. Otherwise there are some cases where tester failures might cause false negative test results.

6.1.3. Scanned window pseudo CBR

Same as the above, except the window is scanned across a range of sizes designed to include two key events, the onset of queueing and the onset of packet loss or ECN marks. The window is scanned by incrementing it by one packet for every $2 \times \text{target_pipe_size}$ delivered packets. This mimics the additive increase phase of standard congestion avoidance and normally separates the the window increases by approximately twice the `target_RTT`.

There are two versions of this test: one built by applying a window clamp to standard congestion control and one one built by stiffening a non-standard transport protocol. When standard congestion control is in effect, any losses or ECN 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 (attempts) to maintain the specified window size independent of losses or ECN marks. Such a stiffened transport explicitly violates mandatory Internet congestion control and is not suitable for in situ testing. It is only appropriate for engineering testing under laboratory conditions. The Windowed Ping tools implemented such a test [WPING]. This tool has been updated and is under test.[mpingSource]

The test procedures in Section 7.2 describe how to the partition the scans into regions and how to interpret the results.

6.1.4. Concurrent or channelized testing

The procedures described in his document are only directly applicable to single stream performance measurement, e.g. one TCP connection. In an Ideal world, we would disallow all performance claims based multiple concurrent stream but this is not practical due to at least two different issues. First, many very high rate link technologies are channelized, and pin individual flows to specific channels to minimize reordering or solve other problems and 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 standard [RFC 5681] and de facto standard [CUBIC] congestion control algorithms have scaling limits, in the sense that as a network over a fixed RTT and MTU gets faster all congestion control algorithms get less accurate. In general their noise immunity drops (a single packet drop should have less effect as individual packets

become smaller relative to the window size) and the control frequency of the AIMD sawtooth also drops, meaning that as TCP is using more total capacity it gets less information about the state of the network and other traffic. These properties are a direct consequence of the original Reno design and are implicitly required by the requirement that all transport protocols be "TCP friendly" [Guidelines] There are a number of reason to want to specify performance in term of multiple concurrent flows. Although there are a number of downsides to @@@@

The use of multiple connections in the Internet has been very controversial since the beginning of the World-Wide-Web[first complaint]. Modern browsers open many connections [BScope]. Experts associated with IETF transport area have frequently spoken against this practice [long list]. It is not inappropriate to assume some small number of concurrent connections (e.g. 4 or 6), to compensate for limitation in TCP. However, choosing too large a number is at risk of being interpreted as a signal by the web browser community that this practice has been embraced by the Internet service provider community. It may not be desirable to send such a signal.

Note that the current proposal for httpbis [SPDY] is specifically designed to work best with a single TCP connection per client server pair, because it uses adaptive compression which requires sending separate compression dictionaries per connection. As long as TCP can use IW10 and some of the transport parameter can be cached, multiple connections provide a negative gain, due to the replicated compression overhead.

The specification to use multiple connections is not recommended for data rates below several Mb/s, which can be attained with run lengths under 10000. Since run length goes as the square of the data rates, at higher rates (see Section 8.3) the run lengths can be unfeasibly large, and multiple connection might be the only feasible approach.

6.1.5. Intermittent Testing

Any test which does not depend on queueing (e.g. the CBR tests) or experiences periodic zero outstanding data during normal operation (e.g. between bursts for the various burst tests), can be formulated as an intermittent test.

The Intermittent testing can be used for ongoing monitoring for changes in subpath quality with minimal disruption users. It should be used in conjunction with the full rate test because this method assesses an average_run_length over a long time interval w.r.t. user sessions. It may false fail due to other legitimate congestion causing traffic or may false pass changes in underlying link

properties (e.g. a modem retraining to an out of contract lower rate).

[Need text about bias (false pass) in the shadow of loss caused by excessive bursts]

6.1.6. Intermittent Scatter Testing

Intermittent scatter testing: when testing the network path to or from an ISP subscriber aggregation point (CMTS, DSLAM, etc), intermittent tests can be spread across a pool of users such that no one users experiences the full impact of the testing, even though the traffic to or from the ISP subscriber aggregation point is sustained at full rate.

6.2. Interpreting the Results

6.2.1. Test outcomes

A singleton is a pass/fail measurement of a subpath. If any subpath fails any test then the end-to-end path is also expected to fail to attain the target performance under some conditions.

In addition we use "inconclusive outcome" to indicate that a test failed to attain the required test conditions. A test is inconclusive if the precomputed traffic pattern was not authentically generated, test preconditions were not met or the measurement results were not statistically significantly.

This is important to the extent that the diagnostic tests use protocols which themselves include built in control systems which might interfere with some aspect of the test. For example consider a test that is implemented by adding rate controls and loss instrumentation to TCP: meeting the run length specification while failing to attain the specified data rate 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 run length measurement. (Note that if the measured run length was too small, 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 vantage independence properties of Model Based Metrics depends on the accuracy of the distinction between conclusive (pass or fail) and inconclusive tests. One way to view inconclusive tests is that they reflect situations where the signature is ambiguous between problems with the the subpath and problems with the diagnostic test itself. One of the goals for evolving diagnostic test designs will be to keep

sharpening this distinction.

One of the goals of evolving the testing process, procedures and measurement point selection should be to minimize the number of inconclusive tests.

Note that procedures that attempt to sweep the target parameter space to find the bounds on some parameter (for example to find the highest data rate for a subpath) are likely to break the location independent properties of Model Based Metrics, because the boundary between passing and inconclusive is extremely likely to be RTT sensitive, because TCP's ability to compensate for problems scales with the number of round trips per second.

6.2.2. Statistical criteria for measuring run_length

When evaluating the observed run_length, we need to determine appropriate packet stream sizes and acceptable error levels for efficient methods of measurement. In practice, can we compare the empirically estimated loss probabilities 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 (loss ratio or other metric, any defect we define).

As each packet is sent and measured, we have an ongoing estimate of the performance in terms of defect to total packet ratio (or an empirical probability). We continue to send until conditions support a conclusion or a maximum sending limit has been reached.

We have a target_defect_probability, 1 defect per target_run_length, where a "defect" is defined as a lost packet, a packet with ECN mark, or other impairment. This constitutes the null Hypothesis:

H0: no more than one defect in target_run_length =
3*(target_pipe_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_defect_probability. Based on analysis of typical values and practical limits on measurement

duration, we choose four times the H_0 probability:

H_1 : one or more defects in $(\text{target_run_length}/4)$ packets

and we can stop sending packets if measurements support rejecting H_0 with the specified Type II error = β (= 0.05 for example), thus preferring the alternate hypothesis H_1 .

H_0 and H_1 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) [StatQC], which also starts with a pair of hypothesis specified as above:

H_0 : p_0 = one defect in target_run_length

H_1 : p_1 = one defect in $\text{target_run_length}/4$

As packets are sent and measurements collected, the tester evaluates the cumulative defect count against two boundaries representing H_0 Acceptance or Rejection (and acceptance of H_1):

Acceptance line: $X_a = -h_1 + sn$

Rejection line: $X_r = h_2 + sn$

where n increases linearly for each packet sent and

$h_1 = \{ \log((1-\alpha)/\beta) \} / k$

$h_2 = \{ \log((1-\beta)/\alpha) \} / k$

$k = \log\{ (p_1(1-p_0)) / (p_0(1-p_1)) \}$

$s = [\log\{ (1-p_0)/(1-p_1) \}] / k$

for p_0 and p_1 as defined in the null and alternative Hypotheses statements above, and α and β as the Type I and Type II error.

The SPRT specifies simple stopping rules:

- o $X_a < \text{defect_count}(n) < X_b$: continue testing
- o $\text{defect_count}(n) \leq X_a$: Accept H_0
- o $\text{defect_count}(n) \geq X_b$: Accept H_1

The calculations above are implemented in the R-tool for Statistical Analysis, in the add-on package for Cross-Validation via Sequential Testing (CVST) [<http://www.r-project.org/>] [Rtool] [CVST] .

Using the equations above, we can calculate the minimum number of packets (n) needed to accept H_0 when x defects are observed. For example, when $x = 0$:

$X_a = 0 = -h_l + s_n$
and $n = h_l / s$

6.2.3. Reordering Tolerance

All tests must be instrumented for reordering [RFC4737].

NB: there is no global consensus for how much reordering tolerance is appropriate or reasonable. ("None" is absolutely unreasonable.)

Section 5 of [RFC4737] proposed a metric that may be sufficient to designate isolated reordered packets as effectively lost, because TCP's retransmission response would be the same.

[As a strawman, we propose the following:] TCP should be able to adapt to reordering as long as the reordering extent is no more than the maximum of one half window or 1 mS, whichever is larger. Note that there is a fundamental tradeoff between tolerance to reordering and how quickly algorithms such as fast retransmit can repair losses. Within this limit on reorder extent, there should be no bound on reordering density.

NB: Traditional TCP implementations were not compatible with this metric, however newer implementations still need to be evaluated

Parameters:

Reordering displacement: the maximum of one half of target_pipe_size or 1 mS.

6.3. Test Qualifications

This entire section might be summarized as "needs to be specified in a FSTDS"

Things to monitor before, during and after a test.

6.3.1. Verify the Traffic Generation Accuracy

[Excess detail for this doc. To be summarized]

for most tests, failing to accurately generate the test traffic indicates an inconclusive tests, since it has to be presumed that the error in traffic generation might have affected the test outcome. To the extent that the network itself had an effect on the the traffic generation (e.g. in the standing queue tests) the possibility exists that allowing too large of error margin in the traffic generation might introduce feedback loops that comprise the vantage independents properties of these tests.

Parameters:

Maximum Data Rate Error The permitted amount that the test traffic can be different than specified for the current test. This is a symmetrical bound.

Maximum Data Rate Overage The permitted amount that the test traffic can be above than specified for the current test.

Maximum Data Rate Underage The permitted amount that the test traffic can be less than specified for the current test.

6.3.2. Verify the absence of cross traffic

[Excess detail for this doc. To be summarized]

The proper treatment of cross traffic is different for different subpaths. In general when testing infrastructure which is associated with only one subscriber, the test should be treated as inconclusive if that subscriber is active on the network. However, for shared infrastructure, the question at hand is likely to be testing if provider has sufficient total capacity. In such cases the presence of cross traffic due to other subscribers is explicitly part of the network conditions and its effects are explicitly part of the test.

@@@ Need to distinguish between ISP managed sharing and unmanaged sharing. e.g. WiFi

Note that canceling tests due to load on subscriber lines may introduce sampling errors for testing other parts of the infrastructure. For this reason tests that are scheduled but not run due to load should be treated as a special case of "inconclusive".

Use a passive packet or SNMP monitoring to verify that the traffic volume on the subpath agrees with the traffic generated by a test. Ideally this should be performed before, during and after each test.

The goal is provide quality assurance on the overall measurement process, and specifically to detect the following measurement failure: a user observes unexpectedly poor application performance, the ISP observes that the access link is running at the rated capacity. Both fail to observe that the user's computer has been infected by a virus which is spewing traffic as fast as it can.

Parameters:

Maximum Cross Traffic Data Rate The amount of excess traffic permitted. Note that this will be different for different tests.

One possible method is an adaptation of: [www-didc.lbl.gov/papers/SCNM-PAM03.pdf](http://www.didc.lbl.gov/papers/SCNM-PAM03.pdf) D Agarwal etal. "An Infrastructure for Passive Network Monitoring of Application Data Streams". Use the same

technique as that paper to trigger the capture of SNMP statistics for the link.

6.3.3. Additional test preconditions

[Excess detail for this doc. To be summarized]

Send pre-load traffic as needed to activate radios with a sleep mode, or other "reactive network" elements (term defined in [draft-morton-ippm-2330-update-01]).

Use the procedure above to confirm that the pre-test background traffic is low enough.

7. Diagnostic Tests

The diagnostic tests are organized by which properties are being tested: run length, standing queues; slowstart bursts; sender rate bursts; and combined tests. The combined tests reduce overhead at the expense of conflating the signatures of multiple failures.

7.1. Basic Data Rate and Run Length Tests

We propose several versions of the basic data rate and run length test. All measure the number of packets delivered between losses or ECN marks, using a data stream that is rate controlled at or below the `target_data_rate`.

The tests below differ in how the data rate is controlled. The data can be paced on a timer, or window controlled at full target data rate. The first two tests implicitly confirm that `sub_path` has sufficient raw capacity to carry the `target_data_rate`. They are recommend for relatively infrequent testing, such as an installation or auditing process. The third, background run length, is a low rate test designed for ongoing monitoring for changes in subpath quality.

All rely on the receiver accumulating packet delivery statistics as described in Section 6.2.2 to score the outcome:

Pass: it is statistically significant that the observed run length is larger than the `target_run_length`.

Fail: it is statistically significant that the observed run length is smaller than the `target_run_length`.

A test is considered to be inconclusive if it failed to meet the data rate as specified below, meet the qualifications defined in

Section 6.3 or neither run length statistical hypothesis was confirmed in the allotted test duration.

7.1.1. Run Length 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 6.1.1 with a burst size of 1 (single packets).

The test is considered to be inconclusive if the packet transmission can not be accurately controlled for any reason.

7.1.2. run length at Full Data Windowed Rate

Confirm that the observed run length is at least the `target_run_length` while sending at an average rate equal to the `target_data_rate`, by controlling (or clamping) the window size of a conventional transport protocol to a fixed value computed from the properties of the test path, typically $\text{test_window} = \text{target_data_rate} * \text{test_RTT} / \text{target_MTU}$.

Since losses and ECN marks generally cause transport protocols to at least temporarily 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. To pass this test the network MUST deliver `target_pipe_size` packets in `target_RTT` time without any losses or ECN marks at least once per two `target_pipe_size` round trips, in addition to meeting the run length statistical test.

7.1.3. Background Run Length 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.

Existing loss metrics such as [RFC 6673] might be appropriate for measuring background run length.

7.2. Standing Queue tests

These test confirm that the bottleneck is well behaved across the onset of packet loss, which typically follows after the onset of

queueing. Well behaved generally means lossless for transient queues, but once the queue has been sustained for a sufficient period of time (or a sufficient queue depth) there should be a small number of losses to signal to the transport protocol that it should reduce its window. 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 be subject to bufferbloat [Bufferbloat] and inflict excess queueing delays on all flows sharing the bottleneck. Excess losses make loss recovery problematic for the transport protocol. Non-linear or erratic RTT fluctuations suggest poor interactions between the channel acquisition systems and the transport self clock. All of the tests in this section use the same basic scanning algorithm but score the link on the basis of how well it avoids each of these problems.

For some technologies the data might not be subject to increasing delays, in which case the data rate will vary with the window size all the way up to the onset of losses or ECN marks. For these technologies, the discussion of queueing does not apply, but it is still required that the onset of losses (or ECN marks) be at an appropriate point and progressive.

Use the procedure in Section 6.1.3 to sweep the window across the onset of queueing 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 $2 \times \text{target_pipe_size}$ packets delivered. A scan can be divided into three regions: below the onset of queueing, a standing queue, and at or beyond the onset of loss.

Below the onset of queueing the RTT is typically fairly constant, and the data rate varies in proportion to the window size. Once the data rate reaches the link rate, the data rate becomes fairly constant, and the RTT increases in proportion to the the window size. The precise transition from one region to the other can be identified by the maximum network power, defined to be the ratio data rate over the $\text{RTT}[\text{POWER}]$.

For technologies that do not have conventional queues, start the scan at a window equal to the test_window, i.e. starting at the target rate, instead of the power point.

If there is random background loss (e.g. bit errors, etc), precise determination of the onset of packet loss may require multiple scans. Above the onset of loss, all transport protocols are expected to experience periodic losses. For the stiffened transport case they will be determined by the AQM algorithm in the network or the details of how the the window increase function responds to loss. For the

standard transport case the details of periodic losses are typically dominated by the behavior of the transport protocol itself.

7.2.1. Congestion Avoidance

A link passes the congestion avoidance standing queue test if more than `target_run_length` packets are delivered between the power point (or `test_window`) and the first loss or ECN mark. If this test is implemented using a standards congestion control algorithm with a clamp, it can be used in situ in the production internet as a capacity test. For an example of such a test see [NPAD].

7.2.2. Bufferbloat

This test confirms that there is some mechanism to limit buffer occupancy (e.g. prevents bufferbloat). Note that this is not strictly a requirement for single stream bulk performance, however if there is no mechanism to limit buffer occupancy then a single stream with sufficient data to deliver is likely to cause the problems described in [RFC 2309] and [Bufferbloat]. This may cause only minor symptoms for the dominant flow, but has the potential to make the link unusable for all other flows and applications.

Pass if the onset of loss is before a standing queue has introduced more delay than twice `target_RTT`, or other well defined 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. Note that in conjunction with the previous test, this test implies that the first loss should occur at a queueing delay which is between one and two times the `target_RTT`.

7.2.3. Non excessive loss

This test confirm that the onset of loss is not excessive. Pass if losses are bound by the the fluctuations in the cross traffic, such that transient load (bursts) do not cause dips in aggregate raw throughput. e.g. pass as long as the losses are no more bursty than are expected from a simple drop tail queue. Although this test could be made more precise it is really included here for pedantic completeness.

7.2.4. Duplex Self Interference

This engineering test confirms a bound on the interactions between the forward data path and the ACK return path. Fail if the RTT rises by more than some fixed bound above the expected queueing time computed from the excess window divided by the link data rate.
@@@ This needs further testing.

7.3. Slowstart tests

These tests mimic slowstart: data is sent at twice the effective bottleneck rate to exercise the queue at the dominant bottleneck.

They are 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. sending data too fast is ok, but sending it slower than twice the actual bottleneck rate as indicated by the ACKs is deemed inconclusive). Space the bursts such that the average data rate is equal to the `target_data_rate`.

7.3.1. Full Window slowstart test

This is a capacity test to confirm that slowstart is not likely to exit prematurely. Send slowstart bursts that are `target_pipe_size` total packets. Accumulate packet delivery statistics as described in Section 6.2.2 to score the outcome. Pass if it is statistically significant that the observed run length is larger than the `target_run_length`. Fail if it is statistically significant that the observed run length is smaller than the `target_run_length`.

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.

7.3.2. Slowstart AQM test

Do a continuous slowstart (send data continuously at `slowstart_rate`), until the first loss, stop, allow the network to drain and repeat, gathering statistics on the last packet delivered before the loss, the loss pattern, maximum 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 and elephants" problem)

This is an engineering test: It would be best performed on a quiescent network or testbed, since cross traffic has the potential to change the results.

7.4. 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 nominally out of scope.

Also, there are a several details that are not precisely defined. For starters there is not a standard server interface rate. 1 Gb/s is very common today, but higher rates (e.g. 10 Gb/s) are becoming cost effective and can be expected to be dominant some time in the future.

Current standards permit TCP to send a full window bursts following an application pause. Congestion Window Validation [RFC 2861], is not required, but even if was it does not take effect until an application pause is longer than an RTO. Since this is standard behavior, it is desirable that the network be able to deliver it, otherwise application pauses will cause unwarranted losses.

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.

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 the network should tolerate server rate bursts. Some bursts must be tolerated by the network, but it is probably unreasonable to expect the network to efficiently deliver all data as a series of bursts.

For this reason, this is the only test for which we explicitly encourage detracting. A TDS should include a table of pairs of derating parameters: what burst size to use as a fraction of the target_pipe_size, and how much each burst size is permitted to reduce the run length, relative to to the target_run_length. @@@@ Needs more work and experimentation.

7.5. Combined Tests

These tests are more efficient from a deployment/operational perspective, but may not be possible to diagnose if they fail.

7.5.1. Sustained burst test

Send target_pipe_size*derate sender interface rate bursts every target_RTT*derate, for derate between 0 and 1. Verify that the observed run length meets target_run_length. Key observations:

- o This test is subpath RTT invariant, as long as the tester can generate the required pattern.
- o The subpath under test is expected to go idle for some fraction of the time: (subpath_data_rate-target_rate)/subpath_data_rate. Failing to do so suggests a problem with the procedure.

- o This test is more strenuous than the slowstart tests: they are not needed if the link passes this test with `derate=1`.
- o A link that passes this test is likely to be able to sustain higher rates (close to `subpath_data_rate`) for paths with RTTs smaller than the `target_RTT`. Offsetting this performance underestimation is part of the rationale behind permitting derating in general.
- o This test can be implemented with standard instrumented TCP[RFC 4898], using a specialized measurement application at one end and a minimal service at the other end [RFC 863, RFC 864]. It may require tweaks to the TCP implementation.
- 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 This test is not totally sufficient: the standing window engineering tests are also needed to be sure that the link is well behaved at and beyond the onset of congestion.
- o This one test can be proven to be the one capacity test to supplant them all.

7.5.2. Live Streaming Media

Model Based Metrics can be implemented as a side effect of serving any non-throughput maximizing traffic, such as streaming media, by applying some additional controls to the traffic. The essential requirement is that the traffic be constrained such that even with arbitrary application pauses, bursts and data rate fluctuations the traffic stays within the envelope determined by all of the individual tests described above, for a specific TDS.

If the serving RTT is less than the `target_RTT`, this constraint is most easily implemented by clamping the transport window size to `test_window=target_data_rate*serving_RTT/target_MTU`. This `test_window` size will limit the both the serving data rate and burst sizes to be no larger than the procedures in Section 7.1.2 and Section 7.4, assuming burst size derating equal to the `serving_RTT` divided by the `target_RTT`.

Note that if the application tolerates fluctuations in its actual data rate (say by use of a playout buffer) it is important that the `target_data_rate` be above the actual average rate needed by the application so it can recover after transient pauses caused by congestion or the application itself. Since the serving RTT is smaller than the `target_RTT`, the worst case bursts that might be generated under these conditions are smaller than called for by Section 7.4

8. Examples

In this section we present TDS for a couple of performance specifications.

Tentatively: 5 Mb/s*50 ms, 1 Mb/s*50ms, 250kbp*100mS

8.1. Near serving HD streaming video

Today the best quality HD video requires slightly less than 5 Mb/s [HDvideo]. Since it is desirable to serve such content locally, we assume that the content will be within 50 mS, which is enough to cover continental Europe or either US coast.

5 Mb/s over a 50 ms path

End to End Parameter	Value	units
target_rate	5	Mb/s
target_RTT	50	ms
target_MTU	1500	bytes
target_pipe_size	22	packets
target_run_length	1452	packets

Table 1

This example uses the most conservative TCP model and no derating.

8.2. Far serving SD streaming video

Standard Quality video typically fits in 1 Mb/s [SDvideo]. This can be reasonably delivered via longer paths with larger. We assume 100mS.

5 Mb/s over a 50 ms path

End to End Parameter	Value	units
target_rate	1	Mb/s
target_RTT	100	ms
target_MTU	1500	bytes
target_pipe_size	9	packets
target_run_length	243	packets

Table 2

This example uses the most conservative TCP model and no derating.

8.3. Bulk delivery of remote scientific data

This example corresponds to 100 Mb/s bulk scientific data over a moderately long RTT. Note that the target_run_length is infeasible for most networks.

100 Mb/s over a 200 ms path

End to End Parameter	Value	units
target_rate	100	Mb/s
target_RTT	200	ms
target_MTU	1500	bytes
target_pipe_size	1741	packets
target_run_length	9093243	packets

Table 3

9. Validation

This document permits alternate models and parameter derating, as described in Section 5.2 and Section 5.3. In exchange for this latitude in the modelling process it requires the ability to demonstrate authentic applications and protocol implementations meeting the target end-to-end performance goals over infrastructure that infinitesimally passes the TDS.

The validation process relies on constructing a test network such that all of the individual load tests pass only infinitesimally, and

proving that an authentic application running over a real TCP implementation (or other protocol as appropriate) can be expected to meet the end-to-end target parameters on such a network.

For example using our example in our HD streaming video TDS described in Section 8.1, the bottleneck data rate should be 5 Mb/s, the per packet random background loss probability should be $1/1453$, for a run length of 1452 packets, the bottleneck queue should be 22 packets and the front path should have just enough buffering to withstand 22 packet line rate bursts. We want every one of the TDS tests to fail if we slightly increase the relevant test parameter, so for example sending a 23 packet slowstart bursts should cause excess (possibly deterministic) packet drops at the dominant queue at the bottleneck. On this infinitesimally passing network it should be possible for a real application using a stock TCP implementation in the vendor's default configuration to attain 5 Mb/s over an 50 ms path.

@@@ Need to better specify the workload: both short and long flows.

The difficult part of this process is arranging for each subpath to infinitesimally pass the individual tests. We suggest two approaches: constraining resources in devices by configuring them not to use all available buffer space or data rate; and preloading subpaths with cross traffic. Note that it is important that a single environment is constructed that infinitesimally 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.

If a TDS validated according to these procedures is used to inform public dialog, the validation experiment itself should also be public with sufficient precision for the experiment to be replicated by other researchers. All components should either be open source or fully specified proprietary implementations that are available to the research community.

TODO: paper proving the validation process.

10. Acknowledgements

Ganga Maguluri suggested the statistical test for measuring loss probability in the target run length.

Meredith Whittaker for improving the clarity of the communications.

11. Informative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.
- [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.
- [I-D.morton-ippm-lmap-path]
Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for LMAP", draft-morton-ippm-lmap-path-00 (work in progress), January 2013.
- [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., "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.
- [BScope] Browserscope, "Browserscope Network tests", Sept 2012,

<<http://www.browserscope.org/?category=network>>.

- [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.
- [StatQC] Montgomery, D., "Introduction to Statistical Quality Control - 2nd ed.", ISBN 0-471-51988-X, 1990.
- [CVST] Krueger, T. and M. Braun, "R package: Fast Cross-Validation via Sequential Testing", version 0.1, 11 2012.
- [LMCUBIC] Ledesma Goyzueta, R. and Y. Chen, "A Deterministic Loss Model Based Analysis of CUBIC, IEEE International Conference on Computing, Networking and Communications (ICNC), E-ISBN : 978-1-4673-5286-4", January 2013.

Appendix A. Model Derivations

The reference `target_run_length` described in Section 5.2 is based on very conservative assumptions: that all window above `target_pipe_size` contributes to a standing queue that raises the RTT, and that classic Reno congestion control is 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 standard, but the section provides offsetting requirements.

These models provide estimates that 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 loss probability spans at least 8 orders of magnitude. 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 might represent a factor of 2 in raw parameter.

Although this document gives a lot of latitude for calculating `target_run_length`, people designing suites of tests need to consider the effect of their choices on the ongoing conversation and tussle about the relevance of "TCP friendliness" as an appropriate model for capacity allocation. Choosing a `target_run_length` that is substantially smaller than the reference `target_run_length` specified in Section 5.2 is equivalent to saying that it is appropriate for the transport research community to abandon "TCP friendliness" as a fairness model and to develop more aggressive Internet transport

protocols, and for applications to continue (or even increase) the number of connections that they open concurrently.

A.1. Aggregate Reno

In Section 5.2 it is assumed that the target rate is the same as the link rate, and any excess window causes a standing queue at the bottleneck. This might be representative of a non-shared access link. An alternative situation would be a heavily aggregated subpath where individual flows do not significantly contribute to the queueing delay, and losses are determined monitoring the average data rate, for example by the use of a virtual queue as in [AFD]. In such a scheme the RTT is constant and TCP's AIMD congestion control causes the data rate to fluctuate in a sawtooth. If the traffic is being controlled in a manner that is consistent with the metrics here, goal would be to make the actual average rate equal to the `target_data_rate`.

We can derive a model for Reno TCP and delayed ACK under the above set of assumptions: for some value of `Wmin`, the window will sweep from `Wmin` to `2*Wmin` in `2*Wmin` RTT. Between losses each sawtooth delivers $(1/2)(Wmin+2*Wmin)(2Wmin)$ packets in `2*Wmin` round trip times. However, unlike the queueing case where `Wmin` = `Target_pipe_size`, we want the average of `Wmin` and `2*Wmin` to be the `target_pipe_size`, so the average rate is the target rate. Thus we want `Wmin` = $(2/3)*target_pipe_size$.

(@@@ something is wrong above) Substituting these together we get:

`target_run_length` = $(8/3)(target_pipe_size^2)$

Note that this is always 88% of the reference run length.

A.2. CUBIC

CUBIC has three operating regions. The model for the expected value of window size derived in [LMCUBIC] assumes operation in the "concave" region only, which is a non-TCP friendly region for long-lived flows. The authors make the following assumptions: packet loss probability, `p`, is independent and periodic, losses occur one at a time, and they are true losses due to tail drop or corruption. This definition of `p` aligns very well with our definition of `target_run_length` and the requirement for progressive loss (AQM).

Although CUBIC window increase depends on continuous time, the authors transform the time to reach the maximum Window size in terms of RTT and a parameter for the multiplicative rate decrease on observing loss, `beta` (whose default value is 0.2 in CUBIC). The

expected value of Window size, $E[W]$, is also dependent on C , a parameter of CUBIC that determines its window-growth aggressiveness (values from 0.01 to 4).

$$E[W] = (C * (RTT/p)^3 * ((4-\beta)/\beta))^{-4}$$

and, further assuming Poisson arrival, the mean throughput, x , is

$$x = E[W]/RTT$$

We note that under these conditions (deterministic single losses), the value of $E[W]$ is always greater than 0.8 of the maximum window size \approx reference_run_length. (as far as I can tell)

Commentary on the consequence of the choice.

Appendix B. Version Control

Formatted: Mon Oct 21 15:42:35 PDT 2013

Authors' Addresses

Matt Mathis
Google, Inc
1600 Amphitheater Parkway
Mountain View, California 93117
USA

Email: 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
URI: <http://home.comcast.net/~acmacm/>

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

A. Morton
AT&T Labs
February 1, 2013

Rate Measurement Test Protocol Problem Statement
draft-ietf-ippm-rate-problem-02

Abstract

There is a rate measurement scenario which has wide-spread attention of Internet access subscribers and seemingly all industry players, including regulators. This memo presents an access rate-measurement problem statement for test protocols to measure IP Performance Metrics. Key test protocol aspects require the ability to control packet size on the tested path and enable asymmetrical packet size testing in a controller-responder architecture.

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 5, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Purpose and Scope	3
3. Active Rate Measurement	5
4. Measurement Method Categories	7
5. Test Protocol Control & Generation Requirements	8
6. Security Considerations	9
7. IANA Considerations	9
8. Acknowledgements	9
9. Appendix	9
10. References	10
10.1. Normative References	10
10.2. Informative References	10
Author's Address	11

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).

The access-rate scenario or use case has 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. This memo

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 characterize the access rate measurement scenario as follows:

- 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 for which methods must also be agreed as a part of the solution.
- o Referring to the reference path defined in [I-D.morton-ippm-lmap-path], possible measurement points include a Subscriber's host (mp000), the access service demarcation point (mp100), Intra IP access where a globally routable address is present (mp150), or the gateway between the measured access network and other networks (mp190).
- o Rates at the edge of the network are several orders of magnitude less than aggregation and core portions.
- o Asymmetrical ingress and egress rates are prevalent.

- o Extremely large scale of access services requires low complexity devices participating at the user end of the path.

Today, the majority of widely deployed access services achieve rates less than 100 Mbit/s, and this is the order of magnitude for which a solution is sought now.

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 actual 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.

Support of active measurement methods will be addressed here, consistent with the IPPM working group's traditional charter. Active measurements require synthetic traffic dedicated to testing, and do not use user traffic.

The actual path used by traffic 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 (the Type-P described in [RFC2330]).

There are several aspects of Type-P where user traffic may be examined and directed to special treatment that may affect transmission rates. The possibilities include:

- o Packet length
- o IP addresses used
- o Transport protocol used (where TCP packets may be routed differently from UDP)

- o Transport Protocol port numbers used

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 the 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 discussion of this is beyond the current scope.

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 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 will mandate that support for one or more categories of rate measurement methods and adequate control features for the methods in the test protocol.

3. Active Rate Measurement

This section lists features of active measurement methods needed to measure access rates in production networks.

Test coordination between source and destination devices through control messages and other basic capabilities described in the methods of IPPM RFCs [RFC2679][RFC2680] are taken as given (these could be listed later, if desired).

Most forms of active testing intrude on user performance to some degree. 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, 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. Obviously, categories of rate measurement methods that use less active test traffic than others with similar accuracy SHALL be preferred for In-Service testing.

On the other hand, Out-of-Service tests where the test path shares no links with In-Service user traffic have none of the congestion or skew concerns, but these tests must address other practical concerns such as conducting measurements within a reasonable time from the tester's point of view. Out-of-Service tests where some part of the test path is shared with In-Service traffic MUST respect the In-Service constraints.

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, it is expected that 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 may be located 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 other end of an end-user access or aggregated leased network segment.

REPORTER:

Use information from test packets and local processes to measure delivered packet rates.

4. Measurement Method Categories

The design of rate measurement methods can be divided into two phases: test stream design and measurement (SENDER and RECEIVER), and a follow-up phase for analysis of the measurement to produce results (REPORTER). The measurement protocol that addresses this problem MUST only serve the test stream generation and measurement functions.

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).
4. One or more packet chirps, where intra-packet spacing typically decreases between adjacent packets in the same chirp and each pair of packets represents a rate for testing purposes.

For all categories, the test protocol MUST support:

1. Variable payload lengths among packet streams
2. Variable length (in packets) among packet streams or ensembles
3. Variable IP header markings among packet streams
4. Choice of UDP transport and variable port numbers, OR, choice of TCP transport and variable port numbers for two-way architectures only, OR BOTH.

5. Variable number of packets-pairs, ensembles, or streams used in a test session

The items above are additional variables that the test protocol MUST be able to identify and control.

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.

>>>>>>

Note: For measurement systems employing TCP 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.mathis-ippm-model-based-metrics]. The general requirement statements needed to describe an "open-loop" TCP sender require some additional discussion.

It may also be useful to specify a control for Bulk Transfer Capacity measurement with fully-specified 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 test packet transferred between SENDER and RECEIVER MUST be compliant with the singleton measurement methods described in IPPM RFCs [RFC2679][RFC2680] (these could be listed later, if desired). The time-stamp information or loss/arrival status for each packet MUST be available for communication to the protocol entity that collects results.

5. Test Protocol Control & Generation Requirements

Essentially, 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 and/or pseudo-one-way test capability in a two-way measurement architecture

The ability to control packet size on the tested path and enable asymmetrical packet size testing in a two-way architecture are REQUIRED.

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].

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. IANA Considerations

This memo makes no requests of IANA.

8. 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 and Marcelo Bagnulo have contributed insightful, clarifying comments that made this a better draft.

9. Appendix

This Appendix was proposed to briefly summarize previous rate measurement experience. (There is a large body of research on rate measurement, so there is a question of what to include and what to omit. Suggestions are welcome.)

10. References

10.1. Normative References

- [RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.
- [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.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, August 2009.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, August 2010.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, October 2010.
- [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.mathis-ippm-model-based-metrics]
Mathis, M., "Model Based Internet Performance Metrics",

draft-mathis-ippm-model-based-metrics-00 (work in progress), October 2012.

[I-D.morton-ippm-lmap-path]

Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for LMAP", draft-morton-ippm-lmap-path-00 (work in progress), January 2013.

[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
URI: <http://home.comcast.net/~acmacm/>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2014

L. Ciavattone
AT&T Labs
R. Geib
Deutsche Telekom
A. Morton
AT&T Labs
M. Wieser
Technical University Darmstadt
October 18, 2013

Test Plan and Results for Advancing RFC 2680 on the Standards Track
draft-ietf-ippm-testplan-rfc2680-04

Abstract

This memo proposes to advance a performance metric RFC along the standards track, specifically RFC 2680 on One-way Loss Metrics. Observing that the metric definitions themselves should be the primary focus rather than the implementations of metrics, this memo describes the test procedures to evaluate specific metric requirement clauses to determine if the requirement has been interpreted and implemented as intended. Two completely independent implementations have been tested against the key specifications of RFC 2680.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 21, 2014.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	4
1.1. RFC 2680 Coverage	5
2. A Definition-centric metric advancement process	5
3. Test configuration	5
4. Error Calibration, RFC 2680	9
4.1. Clock Synchronization Calibration	9
4.2. Packet Loss Determination Error	10
5. Pre-determined Limits on Equivalence	10
6. Tests to evaluate RFC 2680 Specifications	11
6.1. One-way Loss, ADK Sample Comparison	11
6.1.1. 340B/Periodic Cross-imp. results	12
6.1.2. 64B/Periodic Cross-imp. results	13
6.1.3. 64B/Poisson Cross-imp. results	14
6.1.4. Conclusions on the ADK Results for One-way Packet Loss	15
6.2. One-way Loss, Delay threshold	15
6.2.1. NetProbe results for Loss Threshold	16
6.2.2. Perfas Results for Loss Threshold	17
6.2.3. Conclusions for Loss Threshold	17
6.3. One-way Loss with Out-of-Order Arrival	17
6.4. Poisson Sending Process Evaluation	18
6.4.1. NetProbe Results	19
6.4.2. Perfas+ Results	20
6.4.3. Conclusions for Goodness-of-Fit	22
6.5. Implementation of Statistics for One-way Loss	22
7. Conclusions for RFC 2680bis	23
8. Security Considerations	23
9. IANA Considerations	23
10. Acknowledgements	23
11. References	24
11.1. Normative References	24
11.2. Informative References	25
Authors' Addresses	26

1. Introduction

The IETF (IP Performance Metrics working group, IPPM) has considered how to advance their metrics along the standards track since 2001.

The renewed work effort sought to investigate ways in which the measurement variability could be reduced and thereby simplify the problem of comparison for equivalence. As a result, there is consensus (captured in [RFC6576]) that equivalent results from independent implementations of metric specifications are sufficient evidence that the specifications themselves are clear and unambiguous; it is the parallel concept of protocol interoperability for metric specifications. The advancement process either produces confidence that the metric definitions and supporting material are clearly worded and unambiguous, OR, identifies ways in which the metric definitions should be revised to achieve clarity. It is a non-goal to compare the specific implementations themselves.

The process also permits identification of options described in the metric RFC that were not implemented, so that they can be removed from the advancing specification (this is an aspect more typical of protocol advancement along the standards track).

This memo's purpose is to implement the current approach for [RFC2680] and document the results.

In particular, this memo documents consensus on the extent of tolerable errors when assessing equivalence in the results. In discussions, the IPPM working group agreed that test plan and procedures should include the threshold for determining equivalence, and this information should be available in advance of cross-implementation comparisons. This memo includes procedures for same-implementation comparisons to help set the equivalence threshold.

Another aspect of the metric RFC advancement process is the requirement to document the work and results. The procedures of [RFC2026] are expanded in [RFC5657], including sample implementation and interoperability reports. This memo follows the template in [RFC6808] for the report that accompanies the protocol action request submitted to the Area Director, including description of the test set-up, procedures, results for each implementation, and conclusions.

The conclusion reached is that [RFC2680] should be advanced on the Standards Track with modifications. The revised text of RFC 2680bis is ready for review [I-D.morton-ippm-2680-bis], but awaits work-in-progress to update the IPPM Framework [RFC2330]. Therefore, this memo documents the information to support [RFC2680] advancement, and the approval of RFC2680bis is left for future action.

1.1. RFC 2680 Coverage

This plan is intended to cover all critical requirements and sections of [RFC2680].

Note that there are only five instances of the requirement term "MUST" in [RFC2680] outside of the boilerplate and [RFC2119] reference.

Material may be added as it is "discovered" (apparently, not all requirements use requirements language).

2. A Definition-centric metric advancement process

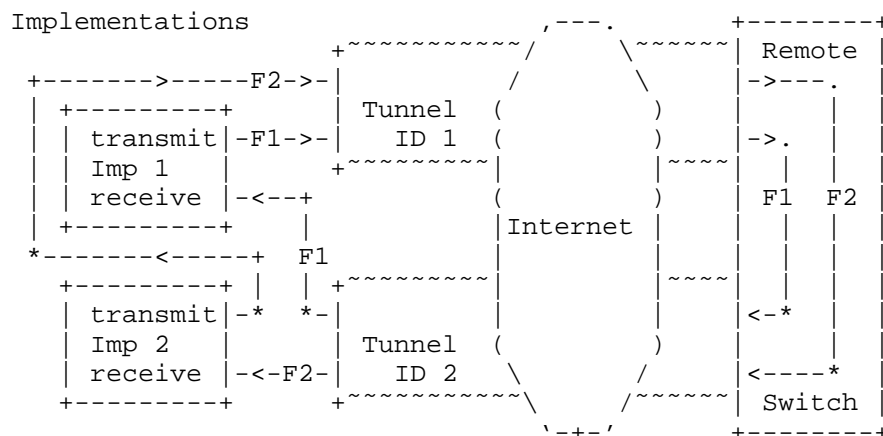
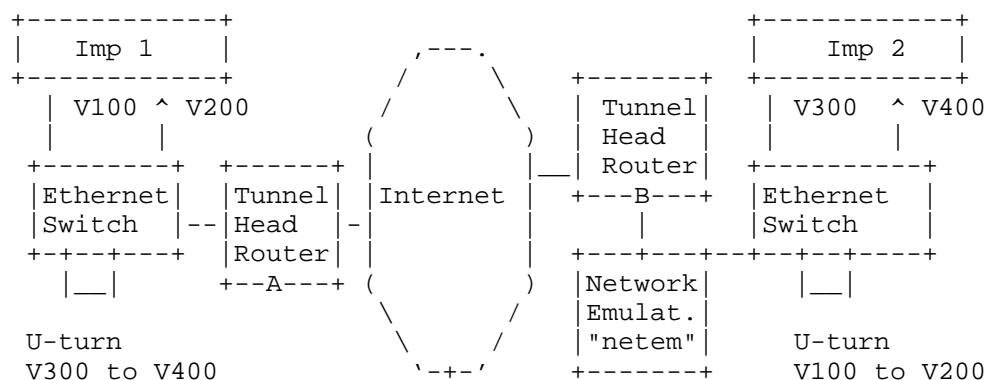
The process described in Section 3.5 of [RFC6576] takes as a first principle that the metric definitions, embodied in the text of the RFCs, are the objects that require evaluation and possible revision in order to advance to the next step on the standards track. This memo follows that process.

3. Test configuration

One metric implementation used was NetProbe version 5.8.5 (an earlier version is used in the WIPM system and deployed world-wide [WIPM]). NetProbe uses UDP packets of variable size, and can produce test streams with Periodic [RFC3432] or Poisson [RFC2330] sample distributions.

The other metric implementation used was Perfas+ version 3.1, developed by Deutsche Telekom [Perfas]. Perfas+ uses UDP unicast packets of variable size (but also supports TCP and multicast). Test streams with periodic, Poisson, or uniform sample distributions may be used.

Figure 1 shows a view of the test path as each Implementation's test flows pass through the Internet and the L2TPv3 tunnel IDs (1 and 2), based on Figure 1 of [RFC6576].



Illustrations of a test setup with a bi-directional tunnel. The upper diagram emphasizes the VLAN connectivity and geographical location (where "Imp #" is the sender and receiver of implementation 1 or 2, either Perfas+ and NetProbe in this test). The lower diagram shows example flows traveling between two measurement implementations. For simplicity only two flows are shown, and netem is omitted (it would appear before or after the Internet, depending on the flow).

Figure 1

The testing employs the Layer 2 Tunnel Protocol, version 3 (L2TPv3) [RFC3931] tunnel between test sites on the Internet. The tunnel IP and L2TPv3 headers are intended to conceal the test equipment addresses and ports from hash functions that would tend to spread different test streams across parallel network resources, with likely

variation in performance as a result.

At each end of the tunnel, one pair of VLANs encapsulated in the tunnel are looped-back so that test traffic is returned to each test site. Thus, test streams traverse the L2TP tunnel twice, but appear to be one-way tests from the test equipment point of view.

The network emulator is a host running Fedora 14 Linux [Fedora] with IP forwarding enabled and the "netem" Network emulator as part of the Fedora Kernel 2.6.35.11 [netem] loaded and operating. The standard kernel is "tickless" replacing the previous periodic timer (250HZ, with 4ms uncertainty) interrupts with on-demand interrupts. Connectivity across the netem/Fedora host was accomplished by bridging Ethernet VLAN interfaces together with "brctl" commands (e.g., eth1.100 <-> eth2.100). The netem emulator was activated on one interface (eth1) and only operates on test streams traveling in one direction. In some tests, independent netem instances operated separately on each VLAN.

The links between the netem emulator host and router and switch were found to be 100baseTx-HD (100Mbps half duplex) as reported by "mii-tool" [mii-tool], when testing was complete. Use of half duplex was not intended, but probably added a small amount of delay variation that could have been avoided in full duplex mode.

Each individual test was run with common packet rates (1 pps, 10pps) Poisson/Periodic distributions, and IP packet sizes of 64, 340, and 500 Bytes.

For these tests, a stream of at least 300 packets was sent from source to destination in each implementation. Periodic streams (as per [RFC3432]) with 1 second spacing were used, except as noted.

As required in Section 2.8.1 of [RFC2680], packet Type-P must be reported. The packet Type-P for this test was IP-UDP with Best Effort DSCP. These headers were encapsulated according to the L2TPv3 specifications [RFC3931], and thus may not influence the treatment received as the packets traversed the Internet.

With the L2TPv3 tunnel in use, the metric name for the testing configured here (with respect to the IP header exposed to Internet processing) is:

Type-IP-protocol-115-One-way-Packet-Loss-<StreamType>-Stream

With (Section 3.2. [RFC2680]) metric parameters:

+ Src, the IP address of a host (12.3.167.16 or 193.159.144.8)

- + Dst, the IP address of a host (193.159.144.8 or 12.3.167.16)
- + T0, a time
- + Tf, a time
- + lambda, a rate in reciprocal seconds
- + Thresh, a maximum waiting time in seconds (see Section 2.8.2 of [RFC2680]) and (Section 3.8. [RFC2680])

Metric Units: A sequence of pairs; the elements of each pair are:

- + T, a time, and
- + L, either a zero or a one

The values of T in the sequence are monotonically increasing. Note that T would be a valid parameter of *singleton* Type-P-One-way-Packet-Loss, and that L would be a valid value of Type-P-One-way-Packet Loss (see Section 2 of [RFC2680]).

Also, Section 2.8.4 of [RFC2680] recommends that the path SHOULD be reported. In this test set-up, most of the path details will be concealed from the implementations by the L2TPv3 tunnels, thus a more informative path trace route can be conducted by the routers at each location.

When NetProbe is used in production, a traceroute is conducted in parallel at the outset of measurements.

Perfas+ does not support traceroute.

```
IPLGW#traceroute 193.159.144.8
```

```
Type escape sequence to abort.
```

```
Tracing the route to 193.159.144.8
```

```

 1 12.126.218.245 [AS 7018] 0 msec 0 msec 4 msec
 2 cr84.n54ny.ip.att.net (12.123.2.158) [AS 7018] 4 msec 4 msec
   cr83.n54ny.ip.att.net (12.123.2.26) [AS 7018] 4 msec
 3 cr1.n54ny.ip.att.net (12.122.105.49) [AS 7018] 4 msec
   cr2.n54ny.ip.att.net (12.122.115.93) [AS 7018] 0 msec
   cr1.n54ny.ip.att.net (12.122.105.49) [AS 7018] 0 msec
 4 n54ny02jt.ip.att.net (12.122.80.225) [AS 7018] 4 msec 0 msec
   n54ny02jt.ip.att.net (12.122.80.237) [AS 7018] 4 msec
 5 192.205.34.182 [AS 7018] 0 msec
   192.205.34.150 [AS 7018] 0 msec
   192.205.34.182 [AS 7018] 4 msec
 6 da-rg12-i.DA.DE.NET.DTAG.DE (62.154.1.30) [AS 3320] 88 msec 88 msec
88 msec
 7 217.89.29.62 [AS 3320] 88 msec 88 msec 88 msec
 8 217.89.29.55 [AS 3320] 88 msec 88 msec 88 msec
 9 * * *
```

NetProbe Traceroute

It was only possible to conduct the traceroute for the measured path on one of the tunnel-head routers (the normal trace facilities of the measurement systems are confounded by the L2TPv3 tunnel encapsulation).

4. Error Calibration, RFC 2680

An implementation is required to report calibration results on clock synchronization in Section 2.8.3 of [RFC2680] (also required in Section 3.7 of [RFC2680] for sample metrics).

Also, it is recommended to report the probability that a packet successfully arriving at the destination network interface is incorrectly designated as lost due to resource exhaustion in Section 2.8.3 of [RFC2680].

4.1. Clock Synchronization Calibration

For NetProbe and Perfas+ clock synchronization test results, refer to Section 4 of [RFC6808].

4.2. Packet Loss Determination Error

Since both measurement implementations have resource limitations, it is theoretically possible that these limits could be exceeded and a packet that arrived at the destination successfully might be discarded in error.

In previous test efforts [I-D.morton-ippm-advance-metrics], NetProbe produced 6 multicast streams with an aggregate bit rate over 53 Mbit/s, in order to characterize the 1-way capacity of a NISTNet-based emulator. Neither the emulator nor the pair of NetProbe implementations used in this testing dropped any packets in these streams.

The maximum load used here between any 2 NetProbe implementations was 11.5 Mbit/s divided equally among 3 unicast test streams. We concluded that steady resource usage does not contribute error (additional loss) to the measurements.

5. Pre-determined Limits on Equivalence

In this section, we provide the numerical limits on comparisons between implementations in order to declare that the results are equivalent and therefore, the tested specification is clear.

A key point is that the allowable errors, corrections, and confidence levels only need to be sufficient to detect misinterpretation of the tested specification resulting in diverging implementations.

Also, the allowable error must be sufficient to compensate for measured path differences. It was simply not possible to measure fully identical paths in the VLAN-loopback test configuration used, and this practical compromise must be taken into account.

For Anderson-Darling K-sample (ADK) [ADK] comparisons, the required confidence factor for the cross-implementation comparisons SHALL be the smallest of:

- o 0.95 confidence factor at 1 packet resolution, or
- o the smallest confidence factor (in combination with resolution) of the two same-implementation comparisons for the same test conditions (if the number of streams is sufficient to allow such comparisons).

For Anderson-Darling Goodness-of-Fit (ADGoF) [Radgof] comparisons, the required level of significance for the same-implementation

Goodness-of-Fit (GoF) SHALL be 0.05 or 5%, as specified in Section 11.4 of [RFC2330]. This is equivalent to a 95% confidence factor.

6. Tests to evaluate RFC 2680 Specifications

This section describes some results from production network (cross-Internet) tests with measurement devices implementing IPPM metrics and a network emulator to create relevant conditions, to determine whether the metric definitions were interpreted consistently by implementors.

The procedures are similar contained in Appendix A.1 of [RFC6576] for One-way Delay.

6.1. One-way Loss, ADK Sample Comparison

This test determines if implementations produce results that appear to come from a common packet loss distribution, as an overall evaluation of Section 3 of [RFC2680], "A Definition for Samples of One-way Packet Loss". Same-implementation comparison results help to set the threshold of equivalence that will be applied to cross-implementation comparisons.

This test is intended to evaluate measurements in sections 2, 3, and 4 of [RFC2680].

By testing the extent to which the counts of one-way packet loss counts on different test streams of two [RFC2680] implementations appear to be from the same loss process, we reduce comparison steps because comparing the resulting summary statistics (as defined in Section 4 of [RFC2680]) would require a redundant set of equivalence evaluations. We can easily check whether the single statistic in Section 4 of [RFC2680] was implemented, and report on that fact.

1. Configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.
2. Measure a sample of one-way packet loss singletons with 2 or more implementations, using identical options and network emulator settings (if used).
3. Measure a sample of one-way packet loss singletons with *four or more* instances of the *same* implementations, using identical options, noting that connectivity differences SHOULD be the same as for cross implementation testing.

4. If less than ten test streams are available, skip to step 7.
5. Apply the ADK comparison procedures (see Appendix C of [RFC6576]) and determine the resolution and confidence factor for distribution equivalence of each same-implementation comparison and each cross-implementation comparison.
6. Take the coarsest resolution and confidence factor for distribution equivalence from the same-implementation pairs, or the limit defined in Section 5 above, as a limit on the equivalence threshold for these experimental conditions.
7. Compare the cross-implementation ADK performance with the equivalence threshold determined in step 5 to determine if equivalence can be declared.

The metric parameters varied for each loss test, and they are listed first in each sub-section below.

The cross-implementation comparison uses a simple ADK analysis [Rtool] [Radk], where all NetProbe loss counts are compared with all Perfas+ loss results.

In the result analysis of this section:

- o All comparisons used 1 packet resolution.
- o No Correction Factors were applied.
- o The 0.95 confidence factor (1.960 for cross-implementation comparison) was used.

6.1.1. 340B/Periodic Cross-imp. results

Tests described in this section used:

- o IP header + payload = 340 octets
- o Periodic sampling at 1 packet per second
- o Test duration = 1200 seconds (during April 7, 2011, EDT)

The netem emulator was set for 100ms constant delay, with 10% loss ratio. In this experiment, the netem emulator was configured to operate independently on each VLAN and thus the emulator itself is a potential source of error when comparing streams that traverse the test path in different directions.

```

=====
A07bps_loss <- c(114, 175, 138, 142, 181, 105) (NetProbe)
A07per_loss <- c(115, 128, 136, 127, 139, 138) (Perfas+)

> A07bps_loss <- c(114, 175, 138, 142, 181, 105)
> A07per_loss <- c(115, 128, 136, 127, 139, 138)
>
> A07cross_loss_ADK <- adk.test(A07bps_loss, A07per_loss)
> A07cross_loss_ADK
Anderson-Darling k-sample test.

```

```

Number of samples: 2
Sample sizes: 6 6
Total number of values: 12
Number of unique values: 11

```

```

Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.6569

```

```

T = (Anderson Darling Criterion - mean)/sigma

```

```

Null Hypothesis: All samples come from a common population.

```

	t.obs	P-value	extrapolation
not adj. for ties	0.52043	0.20604	0
adj. for ties	0.62679	0.18607	0

```

=====

```

```

The cross-implementation comparisons pass the ADK criterion.

```

6.1.2. 64B/Periodic Cross-imp. results

```

Tests described in this section used:

```

- o IP header + payload = 64 octets
- o Periodic sampling at 1 packet per second
- o Test duration = 300 seconds (during March 24, 2011, EDT)

```

The netem emulator was set for 0ms constant delay, with 10% loss
ratio.

```

```
=====
```

```
> M24per_loss <- c(42,34,35,35)          (Perfas+)
> M24apd_23BC_loss <- c(27,39,29,24)      (NetProbe)
> M24apd_loss23BC_ADK <- adk.test(M24apd_23BC_loss,M24per_loss)
> M24apd_loss23BC_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 4 4
Total number of values: 8
Number of unique values: 7
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.60978
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

```
Null Hypothesis: All samples come from a common population.
```

	t.obs	P-value	extrapolation
not adj. for ties	0.76921	0.16200	0
adj. for ties	0.90935	0.14113	0

```
Warning: At least one sample size is less than 5.
p-values may not be very accurate.
```

```
=====
```

```
The cross-implementation comparisons pass the ADK criterion.
```

6.1.3. 64B/Poisson Cross-imp. results

```
Tests described in this section used:
```

- o IP header + payload = 64 octets
- o Poisson sampling at lambda = 1 packet per second
- o Test duration = 20 minutes (during April 27, 2011, EDT)

```
The netem configuration was 0ms delay and 10% loss, but there were
two passes through an emulator for each stream, and loss emulation
was present for 18 minutes of the 20 minute test.
```

```

=====
A27aps_loss <- c(91,110,113,102,111,109,112,113) (NetProbe)
A27per_loss <- c(95,123,126,114) (Perfas+)

A27cross_loss_ADK <- adk.test(A27aps_loss, A27per_loss)

> A27cross_loss_ADK
Anderson-Darling k-sample test.

Number of samples: 2
Sample sizes: 8 4
Total number of values: 12
Number of unique values: 11

Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.65642

T = (Anderson Darling Criterion - mean)/sigma

Null Hypothesis: All samples come from a common population.

      t.obs P-value extrapolation
not adj. for ties 2.15099 0.04145      0
adj. for ties    1.93129 0.05125      0

Warning: At least one sample size is less than 5.
p-values may not be very accurate.
>

=====

The cross-implementation comparisons barely pass the ADK criterion at
95% = 1.960 when adjusting for ties.

```

6.1.4. Conclusions on the ADK Results for One-way Packet Loss

We conclude that the two implementations are capable of producing equivalent one-way packet loss measurements based on their interpretation of [RFC2680].

6.2. One-way Loss, Delay threshold

This test determines if implementations use the same configured maximum waiting time delay from one measurement to another under different delay conditions, and correctly declare packets arriving in excess of the waiting time threshold as lost.

See Section 2.8.2 of [RFC2680].

1. Configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.
2. Configure the network emulator to add 1sec one-way constant delay in one direction of transmission.
3. Measure (average) one-way delay with 2 or more implementations, using identical waiting time thresholds (Thresh) for loss set at 3 seconds.
4. Configure the network emulator to add 3 sec one-way constant delay in one direction of transmission equivalent to 2 seconds of additional one-way delay (or change the path delay while test is in progress, when there are sufficient packets at the first delay setting).
5. Repeat/continue measurements.
6. Observe that the increase measured in step 5 caused all packets with 2 sec additional delay to be declared lost, and that all packets that arrive successfully in step 3 are assigned a valid one-way delay.

The common parameters used for tests in this section are:

- o IP header + payload = 64 octets
- o Poisson sampling at $\lambda = 1$ packet per second
- o Test duration = 900 seconds total (March 21, 2011 EDT)

The netem emulator settings added constant delays as specified in the procedure above.

6.2.1. NetProbe results for Loss Threshold

In NetProbe, the Loss Threshold was implemented uniformly over all packets as a post-processing routine. With the Loss Threshold set at 3 seconds, all packets with one-way delay >3 seconds were marked "Lost" and included in the Lost Packet list with their transmission time (as required in Section 3.3 of [RFC2680]). This resulted in 342 packets designated as lost in one of the test streams (with average delay = 3.091 sec).

6.2.2. Perfas Results for Loss Threshold

Perfas+ uses a fixed Loss Threshold which was not adjustable during this study. The Loss Threshold is approximately one minute, and emulation of a delay of this size was not attempted. However, it is possible to implement any delay threshold desired with a post-processing routine and subsequent analysis. Using this method, 195 packets would be declared lost (with average delay = 3.091 sec).

6.2.3. Conclusions for Loss Threshold

Both implementations assume that any constant delay value desired can be used as the Loss Threshold, since all delays are stored as a pair <Time, Delay> as required in [RFC2680]. This is a simple way to enforce the constant loss threshold envisioned in [RFC2680] (see specific section reference above). We take the position that the assumption of post-processing is compliant, and that the text of the RFC should be revised slightly to include this point.

6.3. One-way Loss with Out-of-Order Arrival

Section 3.6 of [RFC2680] indicates that implementations need to ensure that reordered packets are handled correctly using an uncapitalized "must". In essence, this is an implied requirement because the correct packet must be identified as lost if it fails to arrive before its delay threshold under all circumstances, and reordering is always a possibility on IP network paths. See [RFC4737] for the definition of reordering used in IETF standard-compliant measurements.

Using the procedure of section 6.1, the netem emulator was set to introduce 10% loss, significant delay (2000 ms) and delay variation (1000 ms), which was sufficient to produce packet reordering because each packet's emulated delay is independent from others.

The tests described in this section used:

- o IP header + payload = 64 octets
- o Periodic sampling = 1 packet per second
- o Test duration = 600 seconds (during May 2, 2011, EDT)

```
=====
> Y02aps_loss <- c(53,45,67,55)      (NetProbe)
> Y02per_loss <- c(59,62,67,69)      (Perfas+)
> Y02cross_loss_ADK <- adk.test(Y02aps_loss, Y02per_loss)
> Y02cross_loss_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 4 4
Total number of values: 8
Number of unique values: 7
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.60978
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

```
Null Hypothesis: All samples come from a common population.
```

	t.obs	P-value	extrapolation
not adj. for ties	1.11282	0.11531	0
adj. for ties	1.19571	0.10616	0

```
Warning: At least one sample size is less than 5.
p-values may not be very accurate.
>
```

```
=====
```

The test results indicate that extensive reordering was present. Both implementations capture the extensive delay variation between adjacent packets. In NetProbe, packet arrival order is preserved in the raw measurement files, so an examination of arrival packet sequence numbers also indicates reordering.

Despite extensive continuous packet reordering present in the transmission path, the distributions of loss counts from the two implementations pass the ADK criterion at 95% = 1.960.

6.4. Poisson Sending Process Evaluation

Section 3.7 of [RFC2680] indicates that implementations need to ensure that their sending process is reasonably close to a classic Poisson distribution when used. Much more detail on sample distribution generation and Goodness-of-Fit testing is specified in Section 11.4 of [RFC2330] and the Appendix of [RFC2330].

In this section, each implementation's Poisson distribution is compared with an idealistic version of the distribution available in the base functionality of the R-tool for Statistical Analysis[Rtool], and performed using the Anderson-Darling Goodness-of-Fit test package (ADGofTest) [Radgof]. The Goodness-of-Fit criterion derived from [RFC2330] requires a test statistic value $AD \leq 2.492$ for 5% significance. The Appendix of [RFC2330] also notes that there may be difficulty satisfying the ADGofTest when the sample includes many packets (when 8192 were used, the test always failed, but smaller sets of the stream passed).

Both implementations were configured to produce Poisson distributions with $\lambda = 1$ packet per second, and assign received packet timestamps in the measurement application (above UDP layer, see the calibration results in Section 4 of [RFC6808] for assessment of error).

6.4.1. NetProbe Results

Section 11.4 of [RFC2330] suggests three possible measurement points to evaluate the Poisson distribution. The NetProbe analysis uses "user-level timestamps made just before or after the system call for transmitting the packet".

The statistical summary for two NetProbe streams is below:

```
=====
> summary(a27ms$s1[2:1152])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0100 0.2900 0.6600 0.9846 1.3800 8.6390
> summary(a27ms$s2[2:1152])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.010 0.280 0.670 0.979 1.365 8.829
=====
```

We see that both the Means are near the specified $\lambda = 1$.

The results of ADGoF tests for these two streams is shown below:

```
=====
> ad.test( a27ms$s1[2:101], pexp, 1)
```

Anderson-Darling GoF Test

```
data: a27ms$s1[2:101] and pexp
AD = 0.8908, p-value = 0.4197
alternative hypothesis: NA
```

```
> ad.test( a27ms$s1[2:1001], pexp, 1)
```

Anderson-Darling GoF Test

```
data: a27ms$s1[2:1001] and pexp
AD = 0.9284, p-value = 0.3971
alternative hypothesis: NA
```

```
> ad.test( a27ms$s2[2:101], pexp, 1)
```

Anderson-Darling GoF Test

```
data: a27ms$s2[2:101] and pexp
AD = 0.3597, p-value = 0.8873
alternative hypothesis: NA
```

```
> ad.test( a27ms$s2[2:1001], pexp, 1)
```

Anderson-Darling GoF Test

```
data: a27ms$s2[2:1001] and pexp
AD = 0.6913, p-value = 0.5661
alternative hypothesis: NA
=====
```

We see that both 100 and 1000 packet sets from two different streams (s1 and s2) all passed the $AD \leq 2.492$ criterion.

6.4.2. Perfas+ Results

Section 11.4 of [RFC2330] suggests three possible measurement points to evaluate the Poisson distribution. The Perfas+ analysis uses "wire times for the packets as recorded using a packet filter". However, due to limited access at the Perfas+ side of the test setup, the captures were made after the Perfas+ streams traversed the production network, adding a small amount of unwanted delay variation to the wire times (and possibly error due to packet loss).

The statistical summary for two Perfas+ streams is below:

```
=====
> summary(a27pe$p1)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.004  0.347   0.788   1.054   1.548   4.231
> summary(a27pe$p2)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0010 0.2710  0.7080  0.9696  1.3740  7.1160
=====
```

We see that both the means are near the specified $\lambda = 1$.

The results of ADGoF tests for these two streams is shown below:

```
=====
> ad.test(a27pe$p1, pexp, 1 )

Anderson-Darling GoF Test

data:  a27pe$p1  and  pexp
AD = 1.1364, p-value = 0.2930
alternative hypothesis: NA

> ad.test(a27pe$p2, pexp, 1 )

Anderson-Darling GoF Test

data:  a27pe$p2  and  pexp
AD = 0.5041, p-value = 0.7424
alternative hypothesis: NA

> ad.test(a27pe$p1[1:100], pexp, 1 )

Anderson-Darling GoF Test

data:  a27pe$p1[1:100]  and  pexp
AD = 0.7202, p-value = 0.5419
alternative hypothesis: NA

> ad.test(a27pe$p1[101:193], pexp, 1 )

Anderson-Darling GoF Test

data:  a27pe$p1[101:193]  and  pexp
AD = 1.4046, p-value = 0.201
alternative hypothesis: NA

> ad.test(a27pe$p2[1:100], pexp, 1 )
```

Anderson-Darling GoF Test

```
data: a27pe$p2[1:100] and pexp
AD = 0.4758, p-value = 0.7712
alternative hypothesis: NA

> ad.test(a27pe$p2[101:193], pexp, 1 )
```

Anderson-Darling GoF Test

```
data: a27pe$p2[101:193] and pexp
AD = 0.3381, p-value = 0.9068
alternative hypothesis: NA
```

```
>
```

```
=====
```

We see that both 193, 100, and 93 packet sets from two different streams (p1 and p2) all passed the AD <= 2.492 criterion.

6.4.3. Conclusions for Goodness-of-Fit

Both NetProbe and Perfas+ implementations produce adequate Poisson distributions according to the Anderson-Darling Goodness-of-Fit at the 5% significance (1-alpha = 0.05, or 95% confidence level).

6.5. Implementation of Statistics for One-way Loss

We check which statistics were implemented, and report on those facts, noting that Section 4 of [RFC2680] does not specify the calculations exactly, and gives only some illustrative examples.

	NetProbe	Perfas
4.1. Type-P-One-way-Packet-Loss-Average (this is more commonly referred to as loss ratio)	yes	yes

Implementation of Section 4 Statistics

We note that implementations refer to this metric as a loss ratio, and this is an area for likely revision of the text to make it more consistent with wide-spread usage.

7. Conclusions for RFC 2680bis

This memo concludes that [RFC2680] should be advanced on the standards track, and recommends the following edits to improve the text (which are not deemed significant enough to affect maturity).

- o Revise Type-P-One-way-Packet-Loss-Ave to Type-P-One-way-Delay-Packet-Loss-Ratio .
- o Regarding implementation of the loss delay threshold (section 6.2), the assumption of post-processing is compliant, and the text of RFC 2680bis should be revised slightly to include this point.
- o The IETF has reached consensus on guidance for reporting metrics in [RFC6703], and this memo should be referenced in RFC2680bis to incorporate recent experience where appropriate.

We note that there are at least two Errata on [RFC2680] and these should be processed as part of the editing process.

We recognize the existence of BCP 170 [RFC6390] providing guidelines for development of drafts describing new performance metrics. However, the advancement of [RFC2680] represents fine-tuning of long-standing specifications based on experience that helped to formulate BCP 170, and material that satisfies some of the requirements of [RFC6390] can be found in other RFCs, such as the IPPM Framework [RFC2330]. Thus, no specific changes to address BCP 170 guidelines are recommended for RFC 2680bis.

8. Security Considerations

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

9. IANA Considerations

This memo makes no requests of IANA, and the authors hope that IANA personnel will be able to use their valuable time in other worthwhile pursuits.

10. Acknowledgements

The authors thank Lars Eggert for his continued encouragement to advance the IPPM metrics during his tenure as AD Advisor.

Nicole Kowalski supplied the needed CPE router for the NetProbe side of the test set-up, and graciously managed her testing in spite of issues caused by dual-use of the router. Thanks Nicole!

The "NetProbe Team" also acknowledges many useful discussions on statistical interpretation with Ganga Maguluri.

Constructive comments and helpful reviews where also provided by Bill Cervený, Joachim Fabini, and Ann Cervený.

11. References

11.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [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.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, 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.
- [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.
- [RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.

- [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.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.
- [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.

11.2. Informative References

- [ADK] Scholz, F. and M. Stephens, "K-sample Anderson-Darling Tests of Fit, for Continuous and Discrete cases", University of Washington, Technical Report No. 81, May 1986.
- [Fedora] "<http://fedoraproject.org/>".
- [I-D.morton-ippm-2680-bis] Almes, G., Zekauskas, M., and A. Morton, "A One-Way Loss Metric for IPPM", draft-morton-ippm-2680-bis-01 (work in progress), August 2013.
- [I-D.morton-ippm-advance-metrics] Morton, A., "Lab Test Results for Advancing Metrics on the Standards Track", draft-morton-ippm-advance-metrics-02 (work in progress), October 2010.
- [Perfas] Heidemann, C., "Qualitaet in IP-Netzen Messverfahren", published by ITG Fachgruppe, 2nd meeting 5.2.3 (NGN) http://www.itg523.de/oeffentlich/01nov/Heidemann_QOS_Messverfahren.pdf , November 2001.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [Radgof] Bellosta, C., "ADGofTest: Anderson-Darling Goodness-of-Fit Test. R package version 0.3.", <http://cran.r-project.org/web/packages/ADGofTest/index.html>, December 2011.
- [Radk] Scholz, F., "adk: Anderson-Darling K-Sample Test and

Combinations of Such Tests. R package version 1.0.", , 2008.

- [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.
- [WIPM] "AT&T Global IP Network", <http://ipnetwork.bgtmo.ip.att.net/pws/index.html>, 2012.
- [mii-tool] "<http://man7.org/linux/man-pages/man8/mii-tool.8.html>".
- [netem] "<http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>".

Authors' Addresses

Len Ciavattone
AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
USA

Phone: +1 732 420 1239
Fax:
Email: lencia@att.com
URI:

Ruediger Geib
Deutsche Telekom
Heinrich Hertz Str. 3-7
Darmstadt, 64295
Germany

Phone: +49 6151 58 12747
Email: Ruediger.Geib@telekom.de

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

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

Matthias Wieser
Technical University Darmstadt
Darmstadt,
Germany

Phone:
Email: matthias_michael.wieser@stud.tu-darmstadt.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2014

L. Zheng
S. Aldrin
Huawei Technologies
October 18, 2013

Gap Analysis of IPPM Passive Measurements
draft-zheng-ippm-passive-gap-analysis-00.txt

Abstract

This document performs a gap analysis of the current state of IPPM WG and ongoing work, in terms of passive measurements, according to the new charter of the IPPM WG.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2014.

Copyright Notice

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

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

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Passive Measurements VS Active Measurements	2
3. Gap Analysis for Passive Measurements	3
3.1. Framework for IP Performance Metrics	3
3.2. IP Performance Metrics	4
3.3. Registry	4
4. Future Work for Passive Measurement	5
5. Security Considerations	5
6. IANA Considerations	5
7. Acknowledgements	5
8. References	5
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

1. Introduction

The IPPM working group has been recently re-chartered. According to the new charter, passive measurement and hybrid measurement methods are now included. This document performs a gap analysis of the current status of work in the IPPM WG in terms of passive measurements. Section 2 of the document gives a brief introduction of passive measurement. Section 3 summarizes the current status of the IPPM, and gives an analysis on what is missing or was not considered, for passive measurements, in terms of framework of metrics, measurement of metrics, registry, etc. Section 4 lists the future work required for passive measurements based on the gap analysis. The analysis for hybrid measurements is out of the scope of this document.

2. Passive Measurements VS Active Measurements

Passive and active measurements are two common approaches for monitoring the network. The passive approach measures real traffic and does not increase the traffic on the network for the measurements. This makes it attractive for in-service monitoring, network trouble-shooting and fault location. Since the passive approach may require viewing packets on the network, there can be privacy or security issues. The active approach relies on the capability to inject test packets into the network, but as such it creates extra traffic. The benefit of active measurements is that

they can be run from virtually anywhere in the network. One difficulty, though, is that the discrete nature of active probing limits the resolution of the measurements. There is also evidence of limitations of probe-based packet loss measurement in low-loss environments. Both passive and active measurements have their strengths and should be regarded as complementary.

3. Gap Analysis for Passive Measurements

This section gives an analysis on what is missing for passive measurements in relation to IPPM, in terms of framework of metrics, measurement of metrics and registry.

3.1. Framework for IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group first created a framework for metric development in [RFC2330], which enabled development of many fundamental metrics. [RFC2330] has been updated once by [RFC5835], which describes a detailed framework for composing and aggregating metrics originally defined in [RFC2330].

The ongoing work [I-D.ietf-ippm-2330-update] proposes to update the IPPM Framework with advanced considerations for measurement methodology and testing. It describes new stream parameters for both network characterization and support of application design using IPPM metrics. All the previous work done for IP performance metrics framework and the ongoing update for the framework has the assumption, which is not explicitly stated, that the measurement method of the metrics is active measurement.

The result of this is, while many of the current framework aspects are still applicable to passive measurement, some of them are not applicable. In one example, section 11 of [RFC2330] introduces a separation between three distinct notions: singletons, samples, and statistics, which are not applicable to passive measurements, since the test packet is not required for passive measurements, nor is the sampling.

But there are certainly equivalent concepts in passive measurements. For example, consider using TCP traffic to determine the two-way delay between two hosts. A singleton would be the timing of a single sequence number - acknowledgement pairing, a sample would be a collection of these, and the statistical metric would take the minimum, over a short time interval (in order to reduce or eliminate think-time and delayed-ACK effects). In another example, the concept of a packet of type "P", while still applicable in principle, will have to be specified differently. An updated or new passive framework document is needed, while equivalent concepts need to be carried over as much as possible with passive-friendly definitions.

3.2. IP Performance Metrics

The IPPM WG has defined more than 30 metrics, the most recently published document that defines metrics is [RFC6049]. The commonly used metrics include IPPM Metrics for Measuring Connectivity [RFC2678], One-way Delay Metrics [RFC2679], One-way Packet Loss Metrics [RFC2680], Round-trip Delay Metrics [RFC2681], One-way Loss Pattern Sample Metrics [RFC3357], IP Packet Delay Variation Metric [RFC3393], IPPM Metrics for periodic streams [RFC3432] etc.

All the existing metrics defined follow the framework for IP performance metrics [RFC2330], which has the implicit assumption that the measurement method of the metrics is active measurement. Passive methodologies for existing [RFC2330] based active metrics need to be defined, which would require loosening some of the constraints as well as changes to the guidelines. For example, the measurement methodologies for One-way Delay Metrics [RFC2679] and One-way Packet Loss Metrics [RFC2680] call for, amongst other things, selection of the Src and Dst addresses at the Src host. This will be difficult to achieve for passive measurement.

Careful examination and thorough analysis needs to be made, in order to decide, which aspects of current metrics need to be redefined for passive measurements, and which aspects could be reused by passive measurements as is.

3.3. Registry

[RFC4148] defines an initial registry of the metrics defined in the IPPM WG and the rules to manage the registry. However, [RFC4148] was obsoleted by [RFC6248] because it was "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". This led to the [RFC4148] registry having "very few users, if any".

The ongoing work [I-D.bagnulo-ippm-new-registry-independent] and [I-D.bagnulo-ippm-new-registry] creates, a registry for commonly used metrics, defines the rules for assignments in the new registry and performs initial allocations, respectively. [I-D.bagnulo-ippm-new-registry-independent] proposes one particular registry structure with independent registries for each of the fields involved, while [I-D.bagnulo-ippm-new-registry] explores an alternative structure with a single registry with multiple sub-registries. The metrics for passive measurement should be taken into consideration for both registry structure designs.

4. Future Work for Passive Measurement

Based on the above gap analysis, it could be concluded that the following new work needs to be done in the IPPM working group:

1. Framework for metrics: An passive-friendly updated framework document is needed for passive measurement.
2. Metrics: Careful examination on currently defined metrics, particularly the measurement aspects, needs to be made by the working group. Some metrics need to be updated for passive measurement, some metrics may be reused by passive measurements as is. New metrics may also need to be defined for passive measurement.
3. Registry: The passive measurement should be taken into consideration for the ongoing registry structure design work.

5. Security Considerations

This document does not bring new security issue to IPPM.

6. IANA Considerations

This document makes no request to IANA.

7. Acknowledgements

The authors would like to thank Brain Trammell, Paul Coverdale for their valuable comments.

8. References

8.1. Normative References

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

8.2. Informative References

- [I-D.bagnulo-ippm-new-registry-independent]
Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A registry for commonly used metrics. Independent registries", draft-bagnulo-ippm-new-registry-independent-01 (work in progress), July 2013.
- [I-D.bagnulo-ippm-new-registry]
Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A registry for commonly used metrics", draft-bagnulo-ippm-new-registry-01 (work in progress), July 2013.
- [I-D.ietf-ippm-2330-update]
Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IPPM", draft-ietf-ippm-2330-update-01 (work in progress), October 2013.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, September 1999.
- [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.
- [RFC3357] Koodli, R. and R. Ravikanth, "One-way Loss Pattern Sample Metrics", RFC 3357, August 2002.
- [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.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [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.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.

Authors' Addresses

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Sam K. Aldrin
Huawei Technologies

Email: aldrin.ietf@gmail.com