

INTERNET-DRAFT

N. Elkins
Inside Products
R. Hamilton
Chemical Abstracts Service
M. Ackermann
BCBS Michigan
February 14, 2015

Intended Status: Proposed Standard
Expires: August 2015

IPPM Considerations for the IPv6 PDM Destination Option
draft-elkins-ippm-pdm-option-03

Table of Contents

1	Background	4
1.1	Terminology	4
1.2	End User Quality of Service (QoS)	4
1.3	Need for a Packet Sequence Number	5
1.4	Rationale for proposed solution	5
1.5	PDM Works in Collaboration with Other Headers	5
2	Measurement Information Derived from PDM	6
2.1	Round-Trip Delay	6
2.2	Server Delay	6
3	Performance and Diagnostic Metrics Destination Option Layout	7
3.1	Destination Options Header	7
3.2	Performance and Diagnostic Metrics Destination Option	7
4	Considerations of Timing Representation	10
4.1	Encoding the Delta-Time Values	10
4.2	Timer registers are different on different hardware	10
4.3	Timer Units on Other Systems	11
4.4	Time Base	11
4.5	Timer-value scaling	12
4.6	Limitations with this encoding method	13
4.7	Lack of precision induced by timer value truncation	14
5	PDM Flow - Simple Client Server	15
5.1	Step 1	15
5.2	Step 2	16
5.3	Step 3	16
5.4	Step 4	17
5.5	Step 5	18
6	Other Flows	19
6.1	PDM Flow - One Way Traffic	19
6.2	PDM Flow - Multiple Send Traffic	20
6.3	PDM Flow - Multiple Send with Errors	21
7	Potential Overhead Considerations	22
8	Security Considerations	23

9 IANA Considerations	23
10 References	23
10.1 Normative References	23
10.2 Informative References	24
11 Acknowledgments	24
Authors' Addresses	24

Abstract

To assess performance problems, measurements based on optional sequence numbers and timing may be embedded in each packet. Such measurements may be interpreted in real-time or after the fact. An implementation of the existing IPv6 Destination Options extension header, the Performance and Diagnostic Metrics (PDM) Destination Options extension header has been proposed in a companion document. This document specifies the field limits, calculations, and usage of the PDM in measurement.

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

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

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

1 Background

To assess performance problems, measurements based on optional sequence numbers and timing may be embedded in each packet. Such measurements may be interpreted in real-time or after the fact. An implementation of the existing IPv6 Destination Options extension header, the Performance and Diagnostic Metrics (PDM) Destination Options extension header has been proposed in a companion document. This document specifies the field limits, calculations, and usage of the PDM in measurement.

1.1 Terminology

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

1.2 End User Quality of Service (QoS)

The difference between timing values in the PDM traveling along with the packet will be used to estimate QoS as experienced by an end user device.

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

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

Response time and consistency are not just "nice to have". On many networks, the impact can be financial hardship or endanger human life. In some cities, the emergency police contact system operates over IP, law enforcement uses TCP/IP networks, transactions on our stock exchanges are settled using IP networks. The critical nature of such activities to our daily lives and financial well-being demand a solution.

1.3 Need for a Packet Sequence Number

While performing network diagnostics of an end-to-end connection, it often becomes necessary to find the device along the network path creating problems. Diagnostic data may be collected at multiple places along the path (if possible), or at the source and destination. Then, the diagnostic data corresponding to each packet at different observation points must be matched for proper measurements. A sequence number in each packet provides sufficient basis for the matching process. If need be, the timing fields may be used along with the sequence number to ensure uniqueness.

This method of data collection along the path is of special use to determine where packet loss or packet corruption is happening.

The packet sequence number needs to be unique in the context of the session (5-tuple).

1.4 Rationale for proposed solution

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

Advantages include:

1. Real measure of actual transactions.
2. Independence from transport layer protocols.
3. Ability to span organizational boundaries with consistent instrumentation
4. No time synchronization needed between session partners

The PDM provides the ability to quickly determine if the (latency) problem is in the network or in the server (application). More intermediate measurements may be needed if the host or network discrimination is not sufficient. At the client, TCP/IP stack time vs. applications time may still need to be broken out by client software.

1.5 PDM Works in Collaboration with Other Headers

The purpose of the PDM is not to supplant all the variables present in all other headers but to provide data which is not available or very difficult to get. The way PDM would be used is by a technician (or tool) looking at a packet capture. Within the packet capture,

they would have available to them the layer 2 header, IP header (v6 or v4), TCP, UCP, ICMP, SCTP or other headers. All information would be looked at together to make sense of the packet flow. The technician or processing tool could analyze, report or ignore the data from PDM, as necessary.

For an example of how PDM can help with TCP retransmit problems, please look at section 8.

2 Measurement Information Derived from PDM

Each packet contains information about the sender and receiver. In IP protocol, the identifying information is called a "5-tuple".

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.)

The PDM contains the following metrics:

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

This information, combined with the 5-tuple, allows the measurement of the following metrics:

1. Round-trip delay
2. Server delay

2.1 Round-Trip Delay

Round-trip delay is the end-to-end delay for a packet from a source host to a destination host. This measurement has been defined, and the advantages and disadvantages discussed in "A Round-trip Delay Metric for IPPM" [RFC2681].

2.2 Server Delay

Server delay is the interval between when a packet is received by a device and the first corresponding 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. The stack delay may be related to network performance. If this aggregate time is seen as a problem, and there is a need to make a clear distinction between application processing time and stack delay, including that caused by the network, then more client based measurements are needed.

3 Performance and Diagnostic Metrics Destination Option Layout

3.1 Destination Options Header

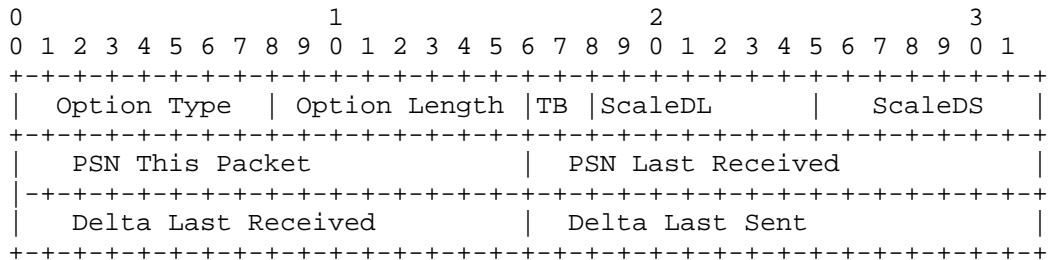
The IPv6 Destination Options Header is used to carry optional information that need be examined only by a packet's destination node(s). The Destination Options Header is identified by a Next Header value of 60 in the immediately preceding header and is defined in RFC2460 [RFC2460]. The IPv6 Performance and Diagnostic Metrics Destination Option (PDM) is an implementation of the Destination Options Header (Next Header value = 60). The PDM does not require time synchronization.

3.2 Performance and Diagnostic Metrics Destination Option

The IPv6 Performance and Diagnostic Metrics Destination Option (PDM) contains the following fields:

- TIMEBASE : Base timer unit
- SCALEDL : Scale for Delta Last Received
- SCALEDL : Scale for Delta Last Sent
- PSNTP : Packet Sequence Number This Packet
- PSNLR : Packet Sequence Number Last Received
- DELTALR : Delta Last Received
- DELTALS : Delta Last Sent

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



Option Type

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

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field MUST be set to 16.

Time Base

2-bit unsigned integer. It will indicate the lowest granularity possible for this device. That is, for a value of 00 in the Time Base field, a value of 1 in the DELTA fields indicates 1 picosecond.

This field is being included so that a device may choose the granularity which most suits its timer ticks. That is, so that it does not have to do more work than needed to convert values required for the PDM.

The possible values of Time Base are as follows:

- 00 - milliseconds
- 01 - microseconds
- 10 - nanoseconds
- 11 - picoseconds

Scale Delta Last Received (SCALEDLR)

7-bit signed integer. This is the scaling value for the Delta Last Received (DELTALR) field. The possible values are from -128 to +127. See Section 4 for further discussion on Timing Considerations and formatting of the scaling values.

Scale Delta Last Sent (SCALEDLS)

7-bit signed integer. This is the scaling value for the Delta Last Sent (DELTALS) field. The possible values are from -128 to +127.

Packet Sequence Number This Packet (PSNTP)

16-bit unsigned integer. This field will wrap. It is intended for human use. That is, while to be used while analyzing packet traces.

Initialized at a random number and monotonically incremented for each

packet on the 5-tuple. The 5-tuple consists of the source and destination IP addresses, the source and destination ports, and the upper layer protocol (ex. TCP, ICMP, etc). The random number initialization is to make it harder to spoof and insert such packets.

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

Packet Sequence Number Last Received (PSNLR)

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

Delta Last Received (DELTALR)

A 16-bit unsigned integer field. The value is according to the scale in SCALEDLR.

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

Delta Last Sent (DELTALS)

A 16-bit unsigned integer field. The value is according to the scale in SCALEDS.

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

Option Type

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

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

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

01 - discard the packet.

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

pointing to the unrecognized Option Type.

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

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

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

0 - Option Data does not change en-route

1 - Option Data may change en-route

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

4 Considerations of Timing Representation

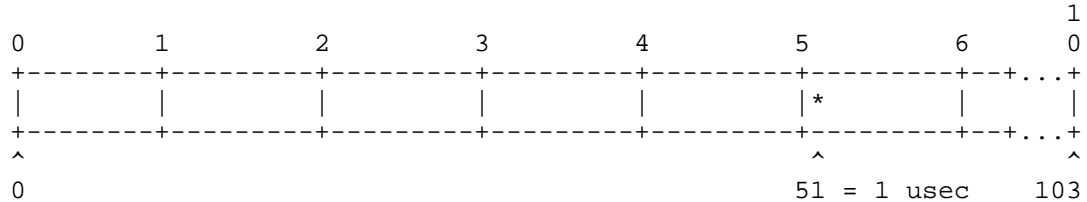
4.1 Encoding the Delta-Time Values

This section makes reference to and expands on the document "Encoding of Time Intervals for the TCP Timestamp Option" [TRAM-TCPM].

4.2 Timer registers are different on different hardware

One of the problems with timestamp recording is the variety of hardware that generates the time value to be used. Different CPUs track the time in registers of different sizes, and the most-frequently-iterated bit could be the first on the left or the first on the right. In order to generate some examples here it is necessary to indicate the type of timer register being used.

As described in the "IBM z/Architecture Principles of Operation" [IBM-POPS], the Time-Of-Day clock in a zSeries CPU is a 104-bit register, where bit 51 is incremented approximately every microsecond:



To represent these values concisely a hexadecimal representation will be used, where each digit represents 4 binary bits. Thus:

```

0000 0000 0000 0001 = 1 timer unit (2**12 usec, or about 244 psec)
0000 0000 0000 1000 = 1 microsecond
0000 0000 003E 8000 = 1 millisecond
0000 0000 F424 0000 = 1 second
0000 0039 3870 0000 = 1 minute
0000 0D69 3A40 0000 = 1 hour
0001 41DD 7600 0000 = 1 day

```

Note that only the first 64 bits of the register are commonly represented, as that represents a count of timer units on this hardware. Commonly the first 52 bits are all that are displayed, as that represents a count of microseconds.

4.3 Timer Units on Other Systems

This encoding method works the same with other hardware clock formats. The method uses a microsecond as the basic value and allows for large time differentials.

4.4 Time Base

We propose a base unit for the time. This is a 2-bit integer indicating the lowest granularity possible for this device. That is, for a value of 00 in the Time Base field, a value of 1 in the DELTA fields indicates 1 picosecond.

The possible values of Time Base are as follows:

```

00 - milliseconds
01 - microseconds
10 - nanoseconds
11 - picoseconds

```

Time base is not necessarily equivalent to length of one timer tick. That is, on many, if not all, systems, the timer tick value will not be in complete units of nanoseconds, milliseconds, etc. For example, on an IBM zSeries machine, one timer tick (or clock unit) is 2 to the -12th microseconds.

Therefore, some amount of conversion may be needed to approximate Time Base units.

4.5 Timer-value scaling

As discussed in [TRAM-TCPM] we propose storing not an entire time-interval value, but just the most significant bits of that value, along with a scaling factor to indicate the magnitude of the time-interval value. In our case, we will use the high-order 16 bits. The scaling value will be the number of bits in the timer register to the right of the 16th significant bit. That is, if the timer register contains this binary value:

```

1110100011010100101001010001000000000000
<-16 bits      -><-24 bits      ->

```

then, the values stored would be 1110 1000 1101 0100 in binary (E8D4 hexadecimal) for the time value and 24 for the scaling value. Note that the displayed value is the binary equivalent of 1 second expressed in picoseconds.

The below table represents a device which has a TimeBase of picosecond (or 00). The smallest and simplest value to represent is 1 picosecond; the time value stored is 1, and the scaling value is 0. Using values from the table below, we have:

Delta time	Time value in picoseconds	Encoded value	Scaling decimal
1 picosecond	1	1	0
1 nanosecond	3E8	3E8	0
1 microsecond	F4240	F424	4
1 millisecond	3B9ACA00	3B9A	16
1 second	E8D4A51000	E8D4	24
1 minute	3691D6AFC000	3691	32
1 hour	cca2e51310000	CCA2	36
1 day	132f4579c980000	132F	44
365 days	1b5a660ea44b80000	1B5A	52

Sample binary values (high order 16 bits taken)

```

1 psec          1                                0001
1 nsec          3E8                                0011 1110 1000
1 usec          F4240                            1111 0100 0010 0100 0000
1 msec          3B9ACA00                        0011 1011 1001 1010 1100 1010 0000 0000
1 sec           E8D4A51000 1110 1000 1101 0100 1010 0101 0001 0000 0000 0000

```

4.6 Limitations with this encoding method

If we follow the specification in [TRAM-TCPM], the size of one of these time-interval fields is limited to this 11-bit value and five-bit scale, so that they fit into a 16-bit space. With that limitation, the maximum value that could be stored in 16 bits is:

```

11-bit value  Scale
=====
1111 1111 111  1 1111

```

or an encoded value of 3FF and a scale value of 31. This value corresponds to any time differential between:

```

|<Count of zeroes is the Scale value>|
11 1111 1111 1000 0000 0000 0000 0000 0000 0000 0000 0000 (binary)
3  F  F  8  0  0  0  0  0  0  0  0  (hexadecimal)

```

and

```

11 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 (binary)
3  F  F  F  F  F  F  F  F  F  F  (hexadecimal)

```

This time value, 3FFFFFFFFF, converts to 50 days, 21 hours, 40 minutes and 46.511103 seconds. A time differential 1 microsecond

longer won't fit into 16 bits using this encoding method.

4.7 Lack of precision induced by timer value truncation

When the bit values following the first 11 significant bits are truncated, obviously loss of precision in the value. The range of values that will be truncated to the same encoded value is $2^{*(Scale)-1}$ microseconds.

The smallest time differential value that will be truncated is

1000 0000 0000 = 2.048 msec

The value

1000 0000 0001 = 2.049 msec

will be truncated to the same encoded value, which is 400 in hex, with a scale value of 1. With the scale value of 1, the value range is calculated as $2^{*1} - 1$, or 1 usec, which you can see is the difference between these minimum and maximum values.

With that in mind, let's look at that table of delta time values again, where the Precision is the range from the smallest value corresponding to this encoded value to the largest:

Delta time	Time value in microseconds	Encoded value	Scale	Precision
1 microsecond	1	1	0	0:00.000000
1 millisecond	38E	38E	0	0:00.000000
1 second	F4240	7A1	9	0:00.000511
1 minute	3938700	727	15	0:00.032767
1 hour	D693A400	6B4	21	0:02.097151
1 day	141DD76000	507	26	1:07.108863
Maximum value	3FFFFFFFFF	7FF	31	35:47.483647

So, when measuring the delay between transmission of two packets, or between the reception of two packets, any delay shorter than 50 days 21 hours and change can be stored in this encoded fashion within 16 bits. When you encode, for example, a DTN response time delay of 50 days, 21 hours and 40 minutes, you can be assured of accuracy within 35 minutes.

5 PDM Flow - Simple Client Server

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

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

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

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

5.1 Step 1

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

The time and packet sequence number are saved by the sender internally. The packet sequence number and delta times are sent in the packet.

Packet 1



PDM Contents:

```
PSNTP   : Packet Sequence Number This Packet:    25
PSNLR   : Packet Sequence Number Last Received:  -
DELTALR : Delta Last Received:                  -
SCALEDL : Scale of Delta LR:                     0
DELTALS : Delta Last Sent:                       -
SCALEDL : Scale of Delta LS:                     0
TIMEBASE : Granularity of Time:                  00 (Picoseconds)
```

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

Packet Sequence Number of the last packet sent: 25
Time the last packet was sent: 10:00:00

Note, the initial PSNTP from Host A starts at a random number. In this case, 25. The timestamp is in seconds for the sake of simplicity.

5.2 Step 2

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

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

Packet Sequence Number of the last packet received: 25
Time the last packet was received : 11:00:03

Note, this timestamp is in Host B time. It has nothing whatsoever to do with Host A time. The Packet Sequence Number of the last packet received will become PSNLR which will be sent out in the packet sent by Host B in the next step. The time last received will be used to calculate the DELTALR value to be sent out in the packet sent by Host B in the next step.

5.3 Step 3

Packet 2 is sent by Host B to Host A. Note, the initial packet sequence number (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.

So, now, Host A can calculate total end-to-end time. That is:

End-to-End Time = 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 time = 10:00:12 - 10:00:00 or 12 (Server and Network RT delay combined)

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

We can now also calculate round trip delay. The formula is:

Round trip delay = DELTALS - DELTALR

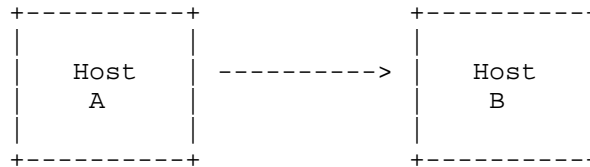
Or:

Round trip delay = 12 - 4 or 8

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

5.5 Step 5

Packet 3 is sent from Host A to Host B.



PDM Contents:

```

PSNTP      : Packet Sequence Number This Packet:    26
PSNLR      : Packet Sequence Number Last Received:  12
DELTALR    : Delta Last Received:                   0
SCALEDL    : Scale of Delta LR                       0
DELTALS    : Delta Last Sent:                        105e (12 seconds)
SCALEDL    : Scale of Delta LR                       26
TIMEBASE   : Granularity of Time:                    00 (Picoseconds)

```

To calculate Two-Way Delay, any packet capture device may look at these packets and do what is necessary.

6 Other Flows

What we have discussed so far is a simple flow with one packet sent and one returned. Let's look at how PDM may be useful in other types of flows.

6.1 PDM Flow - One Way Traffic

The flow on a particular session may not be a send-receive paradigm. Let us consider some other situations. In the case of a one-way flow, one might see the following:

Packet	Sender	PSN This Packet	PSN Last Recvd	Delta Last Recvd	Delta Last Sent
1	Server	1	0	0	0
2	Server	2	0	0	5
3	Server	3	0	0	12
4	Server	4	0	0	20

What does this mean and how is it useful?

In a one-way flow, only the Delta Last Sent will be seen as used. Recall, Delta Last Sent is the difference between the send of one packet from a device and the next. This is a measure of throughput for the sender - according to the sender's point of view. That is, it is a measure of how fast is the application itself (with stack time included) able to send packets.

How might this be useful? If one is having a performance issue at the client and sees that packet 2, for example, is sent after 5 microseconds from the server but takes 3 minutes to arrive at the destination, then one may safely conclude that there are delays in the path other than at the server which may be causing the delivery issue of that packet. Such delays may include the network links, middle-boxes, etc.

Now, true one-way traffic is quite rare. What people often mean by "one-way" traffic is an application such as FTP where a group of packets (for example, a TCP window size worth) is sent, then the sender waits for acknowledgment. This type of flow would actually fall into the "multiple-send" traffic model.

6.2 PDM Flow - Multiple Send Traffic

Assume that two packets are sent with each send from the server.

Packet	Sender	PSN This Packet	PSN Last Recvd	Delta Last Recvd	Delta Last Sent
1	Server	1	0	0	0
2	Server	2	0	0	5
3	Client	1	1	20	0
4	Server	3	1	10	15

How might this be used?

Notice that in packet 3, the client has a value of Delta Last received of 20. Recall that Delta Last Received is the Send time of packet 3 - receive time of packet 2. So, what does one know now? In this case, Delta Last Received is the processing time for the Client to send the next packet.

How to interpret this depends on what is actually being sent. Remember, PDM is not being used in isolation, but to supplement the fields found in other headers. Let's take some examples:

1. Client is sending a standalone TCP ACK. One would find this by looking at the payload length in the IPv6 header and the TCP Acknowledgement field in the TCP header. So, in this case, the client is taking 20 units to send back the ACK. This may or may not be interesting.

2. Client is sending data with the packet. Again, one would find this by looking at the payload length in the IPv6 header and the TCP Acknowledgement field in the TCP header. So, in this case, the client is taking 20 units to send back data. This may represent "User Think Time". Again, this may or may not be interesting, in isolation. But, if there is a performance problem receiving data at the server, then taken in conjunction with RTT or other packet timing information, this information may be quite interesting.

Of course, one also needs to look at the PSN Last Received field to make sure of the interpretation of this data. That is, to make sure that the Delta Last Received corresponds to the packet of interest.

The benefits of PDM are that we have such information available in a uniform manner for all applications and all protocols without extensive changes required to applications.

6.3 PDM Flow - Multiple Send with Errors

One might wonder if all of the functions of PDM might be better suited to TCP or a TCP option. Let us take the case of how PDM may help in a case of TCP retransmissions in a way that TCP options or TCP ACK / SEQ would not.

Assume that three packets are sent with each send from the server.

From the server, this is what is seen.

Pkt	Sender	PSN This Pkt	PSN LastRecvd	Delta LastRecvd	Delta LastSent	TCP SEQ	Data Bytes
1	Server	1	0	0	0	123	100
2	Server	2	0	0	5	223	100
3	Server	3	0	0	5	333	100

The client however, does not get all the packets. From the client, this is what is seen for the packets sent from the server.

Pkt	Sender	PSN This Pkt	PSN LastRecvd	Delta LastRecvd	Delta LastSent	TCP SEQ	Data Bytes
1	Server	1	0	0	0	123	100
2	Server	3	0	0	5	333	100

Let's assume that the server now retransmits the packet. (Obviously, a duplicate acknowledgment sequence for fast retransmit or a retransmit timeout would occur. To illustrate the point, these packets are being left out.)

So, then if a TCP retransmission is done, then from the client, this is what is seen for the packets sent from the server.

Pkt	Sender	PSN This Pkt	PSN LastRecvd	Delta LastRecvd	Delta LastSent	TCP SEQ	Data Bytes
1	Server	4	0	0	30	223	100

The server has resent the old packet 2 with TCP sequence number of 223. The retransmitted packet now has a PSN This Packet value of 4. The Delta Last Sent is 30 - the time between sending the packet with PSN of 3 and this current packet.

Let's say that packet 4 STILL does not make it. Then, after some amount of time (RTO) then the packet with TCP sequence number of 223

is resent.

From the client, this is what is seen for the packets sent from the server.

Pkt	Sender	PSN This Pkt	PSN LastRecvd	Delta LastRecvd	Delta LastSent	TCP SEQ	Data Bytes
1	Server	5	0	0	60	223	100

If now, this packet makes it, one has a very good idea that packets exist which are being sent from the server as retransmissions and not making it to the client. This is because the PSN of the resent packet from the server is 5 rather than 4. If we had used TCP sequence number alone, we would never have seen this situation. Because the TCP sequence number in all situations is 223.

This situation would be experienced by the user of the application (the human being actually sitting somewhere) as a "hangs" or long delay between packets. On large networks, to diagnose problems such as these where packets are lost somewhere on the network, one has to take multiple traces to find out exactly where.

The first thing is to start with doing a trace at the client and the server. So, we can see if the server sent a particular packet and the client received it. If the client did not receive it, then we start tracking back to trace points at the router right after the server and the router right before the client. Did they get these packets which the server has sent? This is a time consuming activity.

With PDM, we can speed up the diagnostic time because we may be able to use only the trace taken at the client to see what the server is sending.

7 Potential Overhead Considerations

Questions have been posed as to the potential overhead of PDM. First, PDM is entirely optional. That is, a site may choose to implement PDM or not as they wish. If they are happy with the costs of PDM vs. the benefits, then the choice should be theirs.

Below is a table outlining the potential overhead in terms of additional time to deliver the response to the end user.

Packet Bytes	RTT	BPM	PDM Bytes	New RTT	Overhead
1000	1000 milli	1	16	1016.000	16.000 milli
1000	100 milli	10	16	101.600	1.600 milli
1000	10 milli	100	16	10.160	.160 milli
1000	1 milli	1000	16	1.016	.016 milli

Below are some examples of actual RTTs for packets traversing large enterprise networks. The first example is for packets going to multiple business partners.

Packet Bytes	RTT	BPM	PDM Bytes	New RTT	Overhead
1000	17 milli	58	16	17.360	.360 milli

The second example is for packets at a large enterprise customer within a data center. Notice that the scale is now in microseconds rather than milliseconds.

Packet Bytes	RTT	BPM	PDM Bytes	New RTT	Overhead
1000	20 micro	50	16	20.320	320 micro

8 Security Considerations

TBD.

9 IANA Considerations

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

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.

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

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

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

[IBM-POPS] IBM Corporation, "IBM z/Architecture Principles of Operation", SA22-7832, 1990-2012

10.2 Informative References

[ELK-PDM] Elkins, N., "draft-elkins-6man-ipv6-pdm-dest-option-09", Internet Draft, October 2014. [Work in Progress]

[TRAM-TCPM] Trammel, B., "Encoding of Time Intervals for the TCP Timestamp Option-01", Internet Draft, July 2013. [Work in Progress]

11 Acknowledgments

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

Authors' Addresses

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

Robert Hamilton
Chemical Abstracts Service
A Division of the American Chemical Society
2540 Olentangy River Road
Columbus, Ohio 43202
United States
Phone: +1 614 447 3600 x2517
Email: rhamilton@cas.org
<http://www.cas.org>

Michael S. Ackermann
Blue Cross Blue Shield of Michigan
P.O. Box 2888
Detroit, Michigan 48231
United States
Phone: +1 310 460 4080
Email: mackermann@bcbsmi.com
<http://www.bcbsmi.com>

Network Working Group
Internet-Draft
Obsoletes: 2679 (if approved)
Intended status: Standards Track
Expires: February 21, 2016

G. Almes
Texas A&M
S. Kalidindi
Ixia
M. Zekauskas
Internet2
A. Morton, Ed.
AT&T Labs
August 20, 2015

A One-Way Delay Metric for IPPM
draft-ietf-ippm-2679-bis-05

Abstract

This memo (RFC 2679 bis) defines a metric for one-way delay of packets across Internet paths. It builds on notions introduced and discussed in the IPPM Framework document, RFC 2330; the reader is assumed to be familiar with that document. This memo makes RFC 2679 obsolete.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 21, 2016.

Copyright Notice

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

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

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

Table of Contents

1.	Changes from RFC 2679	3
2.	Introduction	5
2.1.	Motivation	6
2.2.	General Issues Regarding Time	7
3.	A Singleton Definition for One-way Delay	8
3.1.	Metric Name:	8
3.2.	Metric Parameters:	8
3.3.	Metric Units:	8
3.4.	Definition:	8
3.5.	Discussion:	9
3.6.	Methodologies:	10
3.7.	Errors and Uncertainties:	11
3.7.1.	Errors or uncertainties related to Clocks	11
3.7.2.	Errors or uncertainties related to Wire-time vs Host-time	12
3.7.3.	Calibration	13
3.8.	Reporting the metric:	15
3.8.1.	Type-P	15
3.8.2.	Loss Threshold	16
3.8.3.	Calibration Results	16
3.8.4.	Path	16
4.	A Definition for Samples of One-way Delay	16
4.1.	Metric Name:	17
4.2.	Metric Parameters:	17
4.3.	Metric Units:	17
4.4.	Definition:	18
4.5.	Discussion:	18
4.6.	Methodologies:	19
4.7.	Errors and Uncertainties:	19
4.8.	Reporting the metric:	19
5.	Some Statistics Definitions for One-way Delay	19
5.1.	Type-P-One-way-Delay-Percentile	20
5.2.	Type-P-One-way-Delay-Median	20
5.3.	Type-P-One-way-Delay-Minimum	21
5.4.	Type-P-One-way-Delay-Inverse-Percentile	21
6.	Security Considerations	21
7.	IANA Considerations	22
8.	Acknowledgements	23
9.	References	23
9.1.	Normative References	23

9.2. Informative References 24
 Authors' Addresses 25

1. Changes from RFC 2679

Note: This section's placement currently preserves minimal differences between this memo and RFC 2679. The RFC Editor should place this section in an appropriate place, and remove this note.

The following text constitutes RFC 2769 bis proposed for advancement on the IETF Standards Track. This section tracks the changes from [RFC2679].

[RFC6808] provides the test plan and results supporting [RFC2679] advancement along the standards track, according to the process in [RFC6576]. The conclusions of [RFC6808] list four minor modifications:

1. Section 6.2.3 of [RFC6808] asserts that the assumption of post-processing to enforce a constant waiting time threshold is compliant, and that the text of the RFC should be revised slightly to include this point. The applicability of post-processing was added in the last list item of section 3.6, below.
2. Section 6.5 of [RFC6808] indicates that Type-P-One-way-Delay-Inverse-Percentile statistic has been ignored in both implementations, so it is a candidate for removal or deprecation in RFC2679bis (this small discrepancy does not affect candidacy for advancement). This statistic was deprecated in section 5.4, below.
3. The IETF has reached consensus on guidance for reporting metrics in [RFC6703], and this memo should be referenced in RFC2679bis to incorporate recent experience where appropriate. This reference was added in the last list item of section 3.6, section 3.8, and in section 5 below.
4. There is currently one erratum with status "Held for document update" for [RFC2679], and this minor revision and additional text was incorporated in RFC2679bis in section 5.1, below.

A number of updates to the [RFC2679] text have been implemented in the text below, to reference key IPPM RFCs that were approved after [RFC2679], and to address comments on the IPPM mailing list describing current conditions and experience.

1. Near the end of section 2.1, update of a network example using ATM and clarification of TCP's affect on queue occupation and importance of one-way delay measurement.
2. Explicit inclusion of the maximum waiting time input parameter in section 3.2 and 4.2, reflecting recognition of this parameter in more recent RFCs and ITU-T Recommendation Y.1540.
3. Addition of reference to RFC6703 in the discussion of packet life time and application timeouts in section 3.5.
4. Addition of reference to the default requirement (that packets be standard-formed) from RFC2330 as a new list item in section 3.5.
5. GPS-based NTP experience replaces "to be tested" in section 3.5.
6. Replaced "precedence" with updated terminology (DS Field) in 3.6 and 3.8.1 (with reference).
7. Added parenthetical guidance on minimizing interval between timestamp placement to send time in section 3.6.
8. Section 3.7.2 notes that some current systems perform host time stamping on the network interface hardware.
9. "instrument" replaced by the defined term "host" in sections 3.7.3 and 3.8.3.
10. Added reference to RFC 3432 Periodic sampling alongside Poisson sampling in section 4, and also noting that a truncated Poisson distribution may be needed with modern networks as described in the IPPM Framework update, RFC7312.
11. Add reference to RFC 4737 Reordering metric in the related discussion of section 4.6, Methodologies.
12. Formatting of Example in section 5.1 modified to match the original (issue with conversion to XML in bis version).
13. Clarifying the conclusions on two related points on harm to measurements (recognition of measurement traffic for unexpected priority treatment and attacker traffic which emulates measurement) in section 6, Security Considerations.
14. Expanded and updated the material on Privacy, and added cautions on use of measurements for reconnaissance in section 6, Security Considerations.

Section 5.4.4 of [RFC6390] suggests a common template for performance metrics partially derived from previous IPPM and BMWG RFCs, but also contains some new items. All of the [RFC6390] Normative points are covered, but not quite in the same section names or orientation. Several of the Informative points are covered. Maintaining the familiar outline of IPPM literature has both value and minimizes unnecessary differences between this revised RFC and current/future IPPM RFCs.

The publication of RFC 6921 suggested an area where this memo might need updating. Packet transfer on Faster-Than-Light (FTL) networks could result in negative delays and packet reordering, however both are covered as possibilities in the current text and no revisions are deemed necessary (we also note that this is an April 1st RFC).

2. Introduction

This memo defines a metric for one-way delay of packets across Internet paths. It builds on notions introduced and discussed in the IPPM Framework document, [RFC2330]; the reader is assumed to be familiar with that document, and its recent update [RFC7312].

This memo is intended to be parallel in structure to a companion document for Packet Loss ("A One-way Packet Loss Metric for IPPM") [RFC2680].

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]. Although [RFC2119] was written with protocols in mind, the key words are used in this document for similar reasons. They are used to ensure the results of measurements from two different implementations are comparable, and to note instances when an implementation could perturb the network.

The structure of the memo is as follows:

- + A 'singleton' analytic metric, called Type-P-One-way-Delay, will be introduced to measure a single observation of one-way delay.
- + Using this singleton metric, a 'sample', called Type-P-One-way-Delay-Poisson-Stream, will be introduced to measure a sequence of singleton delays sent at times taken from a Poisson process.
- + Using this sample, several 'statistics' of the sample will be defined and discussed. This progression from singleton to sample to statistics, with clear separation among them, is important.

Whenever a technical term from the IPPM Framework document is first used in this memo, it will be tagged with a trailing asterisk. For example, "term*" indicates that "term" is defined in the Framework.

2.1. Motivation

One-way delay of a Type-P* packet from a source host* to a destination host is useful for several reasons:

- + Some applications do not perform well (or at all) if end-to-end delay between hosts is large relative to some threshold value.
- + Erratic variation in delay makes it difficult (or impossible) to support many real-time applications.
- + The larger the value of delay, the more difficult it is for transport-layer protocols to sustain high bandwidths.
- + The minimum value of this metric provides an indication of the delay due only to propagation and transmission delay.
- + The minimum value of this metric provides an indication of the delay that will likely be experienced when the path* traversed is lightly loaded.
- + Values of this metric above the minimum provide an indication of the congestion present in the path.

The measurement of one-way delay instead of round-trip delay is motivated by the following factors:

- + In today's Internet, the path from a source to a destination may be different than the path from the destination back to the source ("asymmetric paths"), such that different sequences of routers are used for the forward and reverse paths. Therefore round-trip measurements actually measure the performance of two distinct paths together. Measuring each path independently highlights the performance difference between the two paths which may traverse different Internet service providers, and even radically different types of networks (for example, research versus commodity networks, or networks with asymmetric link capacities, or wireless vs. wireline access).
- + Even when the two paths are symmetric, they may have radically different performance characteristics due to asymmetric queueing.
- + Performance of an application may depend mostly on the performance in one direction. For example, a TCP-based communication will

experience reduced throughput if congestion occurs in one direction of its communication. Trouble shooting may be simplified if the congested direction of TCP transmission can be identified.

+ In quality-of-service (QoS) enabled networks, provisioning in one direction may be radically different than provisioning in the reverse direction, and thus the QoS guarantees differ. Measuring the paths independently allows the verification of both guarantees.

It is outside the scope of this document to say precisely how delay metrics would be applied to specific problems.

2.2. General Issues Regarding Time

{Comment: the terminology below differs from that defined by ITU-T documents (e.g., G.810, "Definitions and terminology for synchronization networks" and I.356, "B-ISDN ATM layer cell transfer performance"), but is consistent with the IPPM Framework document. In general, these differences derive from the different backgrounds; the ITU-T documents historically have a telephony origin, while the authors of this document (and the Framework) have a computer systems background. Although the terms defined below have no direct equivalent in the ITU-T definitions, after our definitions we will provide a rough mapping. However, note one potential confusion: our definition of "clock" is the computer operating systems definition denoting a time-of-day clock, while the ITU-T definition of clock denotes a frequency reference.}

Whenever a time (i.e., a moment in history) is mentioned here, it is understood to be measured in seconds (and fractions) relative to UTC.

As described more fully in the Framework document, there are four distinct, but related notions of clock uncertainty:

synchronization*

measures the extent to which two clocks agree on what time it is. For example, the clock on one host might be 5.4 msec ahead of the clock on a second host. {Comment: A rough ITU-T equivalent is "time error".}

accuracy*

measures the extent to which a given clock agrees with UTC. For example, the clock on a host might be 27.1 msec behind UTC. {Comment: A rough ITU-T equivalent is "time error from UTC".}

resolution*

measures the precision of a given clock. For example, the clock on an old Unix host might tick only once every 10 msec, and thus have a resolution of only 10 msec. {Comment: A very rough ITU-T equivalent is "sampling period".}

skew*

measures the change of accuracy, or of synchronization, with time. For example, the clock on a given host might gain 1.3 msec per hour and thus be 27.1 msec behind UTC at one time and only 25.8 msec an hour later. In this case, we say that the clock of the given host has a skew of 1.3 msec per hour relative to UTC, which threatens accuracy. We might also speak of the skew of one clock relative to another clock, which threatens synchronization. {Comment: A rough ITU-T equivalent is "time drift".}

3. A Singleton Definition for One-way Delay

3.1. Metric Name:

Type-P-One-way-Delay

3.2. Metric Parameters:

- + Src, the IP address of a host
- + Dst, the IP address of a host
- + T, a time
- + Tmax, a loss threshold waiting time

3.3. Metric Units:

The value of a Type-P-One-way-Delay is either a real number, or an undefined (informally, infinite) number of seconds.

3.4. Definition:

For a real number dT , \gg the *Type-P-One-way-Delay* from Src to Dst at T is dT \ll means that Src sent the first bit of a Type-P packet to Dst at wire-time* T and that Dst received the last bit of that packet at wire-time $T+dT$.

\gg The *Type-P-One-way-Delay* from Src to Dst at T is undefined (informally, infinite) \ll means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst did not receive that packet (within the loss threshold waiting time, Tmax).

Suggestions for what to report along with metric values appear in Section 3.8 after a discussion of the metric, methodologies for measuring the metric, and error analysis.

3.5. Discussion:

Type-P-One-way-Delay is a relatively simple analytic metric, and one that we believe will afford effective methods of measurement.

The following issues are likely to come up in practice:

- + Real delay values will be positive. Therefore, it does not make sense to report a negative value as a real delay. However, an individual zero or negative delay value might be useful as part of a stream when trying to discover a distribution of a stream of delay values.

- + Since delay values will often be as low as the 100 usec to 10 msec range, it will be important for Src and Dst to synchronize very closely. GPS systems afford one way to achieve synchronization to within several 10s of usec. Ordinary application of NTP may allow synchronization to within several msec, but this depends on the stability and symmetry of delay properties among those NTP agents used, and this delay is what we are trying to measure. A combination of some GPS-based NTP servers and a conservatively designed and deployed set of other NTP servers should yield good results. This was tested in [RFC6808], where a GPS measurement system's results compared well with a GPS-based NTP synchronized system for the same intercontinental path.

- + A given methodology will have to include a way to determine whether a delay value is infinite or whether it is merely very large (and the packet is yet to arrive at Dst). As noted by Mahdavi and Paxson [RFC2678], simple upper bounds (such as the 255 seconds theoretical upper bound on the lifetimes of IP packets [RFC0791]) could be used; but good engineering, including an understanding of packet lifetimes, will be needed in practice. {Comment: Note that, for many applications of these metrics, the harm in treating a large delay as infinite might be zero or very small. A TCP data packet, for example, that arrives only after several multiples of the RTT may as well have been lost. See section 4.1.1 of [RFC6703] for examination of unusual packet delays and application performance estimation.}

- + If the packet is duplicated along the path (or paths) so that multiple non-corrupt copies arrive at the destination, then the packet is counted as received, and the first copy to arrive determines the packet's one-way delay.

- + If the packet is fragmented and if, for whatever reason, reassembly does not occur, then the packet will be deemed lost.

- + The packet is standard-formed, the default criteria for all metric definitions defined in Section 15 of [RFC2330], otherwise the packet will be deemed lost. Note: At this time, the definition of standard-formed packets only applies to IPv4, but also see [I-D.morton-ippm-2330-stdform-typep].

3.6. Methodologies:

As with other Type-P-* metrics, the detailed methodology will depend on the Type-P (e.g., protocol number, UDP/TCP port number, size, Differentiated Services (DS) Field [RFC2780]).

Generally, for a given Type-P, the methodology would proceed as follows:

- + Arrange that Src and Dst are synchronized; that is, that they have clocks that are very closely synchronized with each other and each fairly close to the actual time.

- + At the Src host, select Src and Dst IP addresses, and form a test packet of Type-P with these addresses. Any 'padding' portion of the packet needed only to make the test packet a given size should be filled with randomized bits to avoid a situation in which the measured delay is lower than it would otherwise be, due to compression techniques along the path. Also see section 3.1.2 of [RFC7312].

- + At the Dst host, arrange to receive the packet.

- + At the Src host, place a timestamp in the prepared Type-P packet, and send it towards Dst (ideally minimizing time before sending).

- + If the packet arrives within a reasonable period of time, take a timestamp as soon as possible upon the receipt of the packet. By subtracting the two timestamps, an estimate of one-way delay can be computed. Error analysis of a given implementation of the method must take into account the closeness of synchronization between Src and Dst. If the delay between Src's timestamp and the actual sending of the packet is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis. Similarly, if the delay between the actual receipt of the packet and Dst's timestamp is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis. See the

next section, "Errors and Uncertainties", for a more detailed discussion.

+ If the packet fails to arrive within a reasonable period of time, T_{max} , the one-way delay is taken to be undefined (informally, infinite). Note that the threshold of 'reasonable' is a parameter of the metric. These points are examined in detail in [RFC6703], including analysis preferences to assign undefined delay to packets that fail to arrive with the difficulties emerging from the informal "infinite delay" assignment, and an estimation of an upper bound on waiting time for packets in transit. Further, enforcing a specific constant waiting time on stored singletons of one-way delay is compliant with this specification and may allow the results to serve more than one reporting audience.

Issues such as the packet format, the means by which Dst knows when to expect the test packet, and the means by which Src and Dst are synchronized are outside the scope of this document. {Comment: We plan to document elsewhere our own work in describing such more detailed implementation techniques and we encourage others to as well.}

3.7. Errors and Uncertainties:

The description of any specific measurement method should include an accounting and analysis of various sources of error or uncertainty. The Framework document provides general guidance on this point, but we note here the following specifics related to delay metrics:

+ Errors or uncertainties due to uncertainties in the clocks of the Src and Dst hosts.

+ Errors or uncertainties due to the difference between 'wire time' and 'host time'.

In addition, the loss threshold may affect the results. Each of these are discussed in more detail below, along with a section ("Calibration") on accounting for these errors and uncertainties.

3.7.1. Errors or uncertainties related to Clocks

The uncertainty in a measurement of one-way delay is related, in part, to uncertainties in the clocks of the Src and Dst hosts. In the following, we refer to the clock used to measure when the packet was sent from Src as the source clock, we refer to the clock used to measure when the packet was received by Dst as the destination clock, we refer to the observed time when the packet was sent by the source clock as T_{source} , and the observed time when the packet was received

by the destination clock as T_{dest} . Alluding to the notions of synchronization, accuracy, resolution, and skew mentioned in the Introduction, we note the following:

+ Any error in the synchronization between the source clock and the destination clock will contribute to error in the delay measurement. We say that the source clock and the destination clock have a synchronization error of T_{synch} if the source clock is T_{synch} ahead of the destination clock. Thus, if we know the value of T_{synch} exactly, we could correct for clock synchronization by adding T_{synch} to the uncorrected value of $T_{dest} - T_{source}$.

+ The accuracy of a clock is important only in identifying the time at which a given delay was measured. Accuracy, per se, has no importance to the accuracy of the measurement of delay. When computing delays, we are interested only in the differences between clock values, not the values themselves.

+ The resolution of a clock adds to uncertainty about any time measured with it. Thus, if the source clock has a resolution of 10 msec, then this adds 10 msec of uncertainty to any time value measured with it. We will denote the resolution of the source clock and the destination clock as R_{source} and R_{dest} , respectively.

+ The skew of a clock is not so much an additional issue as it is a realization of the fact that T_{synch} is itself a function of time. Thus, if we attempt to measure or to bound T_{synch} , this needs to be done periodically. Over some periods of time, this function can be approximated as a linear function plus some higher order terms; in these cases, one option is to use knowledge of the linear component to correct the clock. Using this correction, the residual T_{synch} is made smaller, but remains a source of uncertainty that must be accounted for. We use the function $E_{synch}(t)$ to denote an upper bound on the uncertainty in synchronization. Thus, $|T_{synch}(t)| \leq E_{synch}(t)$.

Taking these items together, we note that naive computation $T_{dest} - T_{source}$ will be off by $T_{synch}(t) \pm (R_{source} + R_{dest})$. Using the notion of $E_{synch}(t)$, we note that these clock-related problems introduce a total uncertainty of $E_{synch}(t) + R_{source} + R_{dest}$. This estimate of total clock-related uncertainty should be included in the error/uncertainty analysis of any measurement implementation.

3.7.2. Errors or uncertainties related to Wire-time vs Host-time

As we have defined one-way delay, we would like to measure the time between when the test packet leaves the network interface of Src and when it (completely) arrives at the network interface of Dst, and we

refer to these as "wire times." If the timings are themselves performed by software on Src and Dst, however, then this software can only directly measure the time between when Src grabs a timestamp just prior to sending the test packet and when Dst grabs a timestamp just after having received the test packet, and we refer to these two points as "host times".

We note that some systems perform host time stamping on the network interface hardware, in an attempt to minimize the difference from wire times.

To the extent that the difference between wire time and host time is accurately known, this knowledge can be used to correct for host time measurements, and the corrected value more accurately estimates the desired (wire time) metric.

To the extent, however, that the difference between wire time and host time is uncertain, this uncertainty must be accounted for in an analysis of a given measurement method. We denote by Hsource an upper bound on the uncertainty in the difference between wire time and host time on the Src host, and similarly define Hdest for the Dst host. We then note that these problems introduce a total uncertainty of Hsource+Hdest. This estimate of total wire-vs-host uncertainty should be included in the error/uncertainty analysis of any measurement implementation.

3.7.3. Calibration

Generally, the measured values can be decomposed as follows:

$$\text{measured value} = \text{true value} + \text{systematic error} + \text{random error}$$

If the systematic error (the constant bias in measured values) can be determined, it can be compensated for in the reported results.

$$\text{reported value} = \text{measured value} - \text{systematic error}$$

therefore

$$\text{reported value} = \text{true value} + \text{random error}$$

The goal of calibration is to determine the systematic and random error generated by the hosts themselves in as much detail as possible. At a minimum, a bound ("e") should be found such that the reported value is in the range (true value - e) to (true value + e) at least 95 percent of the time. We call "e" the calibration error for the measurements. It represents the degree to which the values produced by the measurement host are repeatable; that is, how closely

an actual delay of 30 ms is reported as 30 ms. {Comment: 95 percent was chosen because (1) some confidence level is desirable to be able to remove outliers, which will be found in measuring any physical property; (2) a particular confidence level should be specified so that the results of independent implementations can be compared; and (3) even with a prototype user-level implementation, 95% was loose enough to exclude outliers.}

From the discussion in the previous two sections, the error in measurements could be bounded by determining all the individual uncertainties, and adding them together to form

$$E_{\text{synch}}(t) + R_{\text{source}} + R_{\text{dest}} + H_{\text{source}} + H_{\text{dest}}.$$

However, reasonable bounds on both the clock-related uncertainty captured by the first three terms and the host-related uncertainty captured by the last two terms should be possible by careful design techniques and calibrating the hosts using a known, isolated, network in a lab.

For example, the clock-related uncertainties are greatly reduced through the use of a GPS time source. The sum of $E_{\text{synch}}(t) + R_{\text{source}} + R_{\text{dest}}$ is small, and is also bounded for the duration of the measurement because of the global time source.

The host-related uncertainties, $H_{\text{source}} + H_{\text{dest}}$, could be bounded by connecting two hosts back-to-back with a high-speed serial link or isolated LAN segment. In this case, repeated measurements are measuring the same one-way delay.

If the test packets are small, such a network connection has a minimal delay that may be approximated by zero. The measured delay therefore contains only systematic and random error in the measurement hosts. The "average value" of repeated measurements is the systematic error, and the variation is the random error.

One way to compute the systematic error, and the random error to a 95% confidence is to repeat the experiment many times - at least hundreds of tests. The systematic error would then be the median. The random error could then be found by removing the systematic error from the measured values. The 95% confidence interval would be the range from the 2.5th percentile to the 97.5th percentile of these deviations from the true value. The calibration error "e" could then be taken to be the largest absolute value of these two numbers, plus the clock-related uncertainty. {Comment: as described, this bound is relatively loose since the uncertainties are added, and the absolute value of the largest deviation is used. As long as the resulting value is not a significant fraction of the measured values, it is a

reasonable bound. If the resulting value is a significant fraction of the measured values, then more exact methods will be needed to compute the calibration error.}

Note that random error is a function of measurement load. For example, if many paths will be measured by one host, this might increase interrupts, process scheduling, and disk I/O (for example, recording the measurements), all of which may increase the random error in measured singletons. Therefore, in addition to minimal load measurements to find the systematic error, calibration measurements should be performed with the same measurement load that the hosts will see in the field.

We wish to reiterate that this statistical treatment refers to the calibration of the host; it is used to "calibrate the meter stick" and say how well the meter stick reflects reality.

In addition to calibrating the hosts for finite one-way delay, two checks should be made to ensure that packets reported as losses were really lost. First, the threshold for loss should be verified. In particular, ensure the "reasonable" threshold is reasonable: that it is very unlikely a packet will arrive after the threshold value, and therefore the number of packets lost over an interval is not sensitive to the error bound on measurements. Second, consider the possibility that a packet arrives at the network interface, but is lost due to congestion on that interface or to other resource exhaustion (e.g. buffers) in the host.

3.8. Reporting the metric:

The calibration and context in which the metric is measured MUST be carefully considered, and SHOULD always be reported along with metric results. We now present four items to consider: the Type-P of test packets, the threshold of infinite delay (if any), error calibration, and the path traversed by the test packets. This list is not exhaustive; any additional information that could be useful in interpreting applications of the metrics should also be reported (see [RFC6703] for extensive discussion of reporting considerations for different audiences).

3.8.1. Type-P

As noted in the Framework document, section 13 of [RFC2330], the value of the metric may depend on the type of IP packets used to make the measurement, or "Type-P". The value of Type-P-One-way-Delay could change if the protocol (UDP or TCP), port number, size, or arrangement for special treatment (e.g., IP DS Field [RFC2780], ECN [RFC3168], or RSVP) changes. Additional packet distinctions

identified in future extensions of the Type-P definition will apply. The exact Type-P used to make the measurements MUST be accurately reported.

3.8.2. Loss Threshold

In addition, the threshold (or methodology to distinguish) between a large finite delay and loss MUST be reported.

3.8.3. Calibration Results

+ If the systematic error can be determined, it SHOULD be removed from the measured values.

+ You SHOULD also report the calibration error, e , such that the true value is the reported value plus or minus e , with 95% confidence (see the last section.)

+ If possible, the conditions under which a test packet with finite delay is reported as lost due to resource exhaustion on the measurement host SHOULD be reported.

3.8.4. Path

Finally, the path traversed by the packet SHOULD be reported, if possible. In general it is impractical to know the precise path a given packet takes through the network. The precise path may be known for certain Type-P on short or stable paths. If Type-P includes the record route (or loose-source route) option in the IP header, and the path is short enough, and all routers* on the path support record (or loose-source) route, then the path will be precisely recorded. This is impractical because the route must be short enough, many routers do not support (or are not configured for) record route, and use of this feature would often artificially worsen the performance observed by removing the packet from common-case processing. However, partial information is still valuable context. For example, if a host can choose between two links* (and hence two separate routes from Src to Dst), then the initial link used is valuable context. {Comment: For example, with Merit's NetNow setup, a Src on one NAP can reach a Dst on another NAP by either of several different backbone networks.}

4. A Definition for Samples of One-way Delay

Given the singleton metric Type-P-One-way-Delay, we now define one particular sample of such singletons. The idea of the sample is to select a particular binding of the parameters Src, Dst, and Type-P, then define a sample of values of parameter T. The means for

defining the values of T is to select a beginning time T_0 , a final time T_f , and an average rate λ , then define a pseudo-random Poisson process of rate λ , whose values fall between T_0 and T_f . The time interval between successive values of T will then average $1/\lambda$.

Note that Poisson sampling is only one way of defining a sample. Poisson has the advantage of limiting bias, but other methods of sampling will be appropriate for different situations. For example, a truncated Poisson distribution may be needed to avoid reactive network state changes during intervals of inactivity, see section 4.6 of [RFC7312]. Sometimes, the goal is sampling with a known bias, and [RFC3432] describes a method for periodic sampling with random start times.

4.1. Metric Name:

Type-P-One-way-Delay-Poisson-Stream

4.2. Metric Parameters:

- + Src, the IP address of a host
- + Dst, the IP address of a host
- + T_0 , a time
- + T_f , a time
- + Tmax, a loss threshold waiting time
- + λ , a rate in reciprocal seconds (or parameters for another distribution)

4.3. Metric Units:

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

- + T , a time, and
- + dT , either a real number or an undefined number of seconds.

The values of T in the sequence are monotonic increasing. Note that T would be a valid parameter to Type-P-One-way-Delay, and that dT would be a valid value of Type-P-One-way-Delay.

4.4. Definition:

Given T_0 , T_f , and λ , we compute a pseudo-random Poisson process beginning at or before T_0 , with average arrival rate λ , and ending at or after T_f . Those time values greater than or equal to T_0 and less than or equal to T_f are then selected. At each of the times in this process, we obtain the value of Type-P-One-way-Delay at this time. The value of the sample is the sequence made up of the resulting $\langle \text{time}, \text{delay} \rangle$ pairs. If there are no such pairs, the sequence is of length zero and the sample is said to be empty.

4.5. Discussion:

The reader should be familiar with the in-depth discussion of Poisson sampling in the Framework document [RFC2330], which includes methods to compute and verify the pseudo-random Poisson process.

We specifically do not constrain the value of λ , except to note the extremes. If the rate is too large, then the measurement traffic will perturb the network, and itself cause congestion. If the rate is too small, then you might not capture interesting network behavior. {Comment: We expect to document our experiences with, and suggestions for, λ elsewhere, culminating in a "best current practices" document.}

Since a pseudo-random number sequence is employed, the sequence of times, and hence the value of the sample, is not fully specified. Pseudo-random number generators of good quality will be needed to achieve the desired qualities.

The sample is defined in terms of a Poisson process both to avoid the effects of self-synchronization and also capture a sample that is statistically as unbiased as possible. {Comment: there is, of course, no claim that real Internet traffic arrives according to a Poisson arrival process.} The Poisson process is used to schedule the delay measurements. The test packets will generally not arrive at Dst according to a Poisson distribution, since they are influenced by the network.

All the singleton Type-P-One-way-Delay metrics in the sequence will have the same values of Src, Dst, and Type-P.

Note also that, given one sample that runs from T_0 to T_f , and given new time values T_0' and T_f' such that $T_0 \leq T_0' \leq T_f' \leq T_f$, the subsequence of the given sample whose time values fall between T_0' and T_f' are also a valid Type-P-One-way-Delay-Poisson-Stream sample.

4.6. Methodologies:

The methodologies follow directly from:

- + the selection of specific times, using the specified Poisson arrival process, and
- + the methodologies discussion already given for the singleton Type-P-One-way-Delay metric.

Care must, of course, be given to correctly handle out-of-order arrival of test packets; it is possible that the Src could send one test packet at TS[i], then send a second one (later) at TS[i+1], while the Dst could receive the second test packet at TR[i+1], and then receive the first one (later) at TR[i]. Metrics for reordering may be found in [RFC4737].

4.7. Errors and Uncertainties:

In addition to sources of errors and uncertainties associated with methods employed to measure the singleton values that make up the sample, care must be given to analyze the accuracy of the Poisson process with respect to the wire-times of the sending of the test packets. Problems with this process could be caused by several things, including problems with the pseudo-random number techniques used to generate the Poisson arrival process, or with jitter in the value of Hsource (mentioned above as uncertainty in the singleton delay metric). The Framework document shows how to use the Anderson-Darling test to verify the accuracy of a Poisson process over small time frames. {Comment: The goal is to ensure that test packets are sent "close enough" to a Poisson schedule, and avoid periodic behavior.}

4.8. Reporting the metric:

You MUST report the calibration and context for the underlying singletons along with the stream. (See "Reporting the metric" for Type-P-One-way-Delay.)

5. Some Statistics Definitions for One-way Delay

Given the sample metric Type-P-One-way-Delay-Poisson-Stream, we now offer several statistics of that sample. These statistics are offered mostly to illustrate what could be done. See [RFC6703] for additional discussion of statistics that are relevant to different audiences.

5.1. Type-P-One-way-Delay-Percentile

Given a Type-P-One-way-Delay-Poisson-Stream and a percent X between 0% and 100%, the Xth percentile of all the dT values in the Stream. In computing this percentile, undefined values are treated as infinitely large. Note that this means that the percentile could thus be undefined (informally, infinite). In addition, the Type-P-One-way-Delay-Percentile is undefined if the sample is empty.

Example: suppose we take a sample and the results are:

```
Stream1 = <
<T1, 100 msec>
<T2, 110 msec>
<T3, undefined>
<T4, 90 msec>
<T5, 500 msec>
>
```

Then the 50th percentile would be 110 msec, since 90 msec and 100 msec are smaller and 500 msec and 'undefined' are larger. See Section 11.3 of [RFC2330] for computing percentiles.

Note that if the possibility that a packet with finite delay is reported as lost is significant, then a high percentile (90th or 95th) might be reported as infinite instead of finite.

5.2. Type-P-One-way-Delay-Median

Given a Type-P-One-way-Delay-Poisson-Stream, the median of all the dT values in the Stream. In computing the median, undefined values are treated as infinitely large. As with Type-P-One-way-Delay-Percentile, Type-P-One-way-Delay-Median is undefined if the sample is empty.

As noted in the Framework document, the median differs from the 50th percentile only when the sample contains an even number of values, in which case the mean of the two central values is used.

Example: suppose we take a sample and the results are:

```
Stream2 = <
```

```
<T1, 100 msec>  
<T2, 110 msec>  
<T3, undefined>  
<T4, 90 msec>  
>
```

Then the median would be 105 msec, the mean of 100 msec and 110 msec, the two central values.

5.3. Type-P-One-way-Delay-Minimum

Given a Type-P-One-way-Delay-Poisson-Stream, the minimum of all the dT values in the Stream. In computing this, undefined values are treated as infinitely large. Note that this means that the minimum could thus be undefined (informally, infinite) if all the dT values are undefined. In addition, the Type-P-One-way-Delay-Minimum is undefined if the sample is empty.

In the above example, the minimum would be 90 msec.

5.4. Type-P-One-way-Delay-Inverse-Percentile

Note: This statistic is deprecated in this version of the memo because of lack of use.

Given a Type-P-One-way-Delay-Poisson-Stream and a time duration threshold, the fraction of all the dT values in the Stream less than or equal to the threshold. The result could be as low as 0% (if all the dT values exceed threshold) or as high as 100%. Type-P-One-way-Delay-Inverse-Percentile is undefined if the sample is empty.

In the above example, the Inverse-Percentile of 103 msec would be 50%.

6. Security Considerations

Conducting Internet measurements raises both security and privacy concerns. This memo does not specify an implementation of the metrics, so it does not directly affect the security of the Internet nor of applications which run on the Internet. However, implementations of these metrics must be mindful of security and privacy concerns.

There are two types of security concerns: potential harm caused by the measurements, and potential harm to the measurements. The measurements could cause harm because they are active, and inject packets into the network. The measurement parameters MUST be carefully selected so that the measurements inject trivial amounts of additional traffic into the networks they measure. If they inject "too much" traffic, they can skew the results of the measurement, and in extreme cases cause congestion and denial of service.

The measurements themselves could be harmed by routers giving measurement traffic a different priority than "normal" traffic, or by an attacker injecting artificial measurement traffic. If routers can recognize measurement traffic and treat it separately, the measurements will not reflect actual user traffic. Therefore, the measurement methodologies SHOULD include appropriate techniques to reduce the probability measurement traffic can be distinguished from "normal" traffic.

If an attacker injects packets emulating traffic that are accepted as legitimate, the loss ratio or other measured values could be corrupted. Authentication techniques, such as digital signatures, may be used where appropriate to guard against injected traffic attacks.

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

Collecting measurements or using measurement results for reconnaissance to assist in subsequent system attacks is quite common. Access to measurement results, or control of the measurement systems to perform reconnaissance should be guarded against. See Section 7 of [I-D.ietf-lmap-framework] (security considerations of the LMAP Framework) for system requirements that help to avoid measurement system compromise.

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

For [RFC2679], special thanks are due to Vern Paxson of Lawrence Berkeley Labs for his helpful comments on issues of clock uncertainty and statistics. Thanks also to Garry Couch, Will Leland, Andy Scherrer, Sean Shapira, and Roland Wittig for several useful suggestions.

For RFC 2679 bis, thanks to Joachim Fabini, Ruediger Geib, Nalini Elkins, and Barry Constantine for sharing their measurement experience as part of their careful reviews. Brian Carpenter and Scott Bradner provided useful feedback at IETF Last Call.

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<http://www.rfc-editor.org/info/rfc2330>>.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <<http://www.rfc-editor.org/info/rfc2678>>.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, DOI 10.17487/RFC2679, September 1999, <<http://www.rfc-editor.org/info/rfc2679>>.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, DOI 10.17487/RFC2680, September 1999, <<http://www.rfc-editor.org/info/rfc2680>>.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, DOI 10.17487/RFC2780, March 2000, <<http://www.rfc-editor.org/info/rfc2780>>.

- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, DOI 10.17487/RFC3432, November 2002, <<http://www.rfc-editor.org/info/rfc3432>>.
- [RFC6576] Geib, R., Ed., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, DOI 10.17487/RFC6576, March 2012, <<http://www.rfc-editor.org/info/rfc6576>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<http://www.rfc-editor.org/info/rfc7312>>.

9.2. Informative References

- [I-D.ietf-lmap-framework]
Eardley, P., Morton, A., Bagnulo, M., Burbidge, T., Aitken, P., and A. Akhter, "A framework for Large-Scale Measurement of Broadband Performance (LMAP)", draft-ietf-lmap-framework-14 (work in progress), April 2015.
- [I-D.morton-ippm-2330-stdform-typep]
Morton, A., Fabini, J., Elkins, N., Ackermann, M., and V. Hegde, "Updates for IPPM's Active Metric Framework: Packets of Type-P and Standard-Formed Packets", draft-morton-ippm-2330-stdform-typep-00 (work in progress), August 2015.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<http://www.rfc-editor.org/info/rfc4737>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<http://www.rfc-editor.org/info/rfc6390>>.

- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, DOI 10.17487/RFC6703, August 2012, <<http://www.rfc-editor.org/info/rfc6703>>.
- [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, DOI 10.17487/RFC6808, December 2012, <<http://www.rfc-editor.org/info/rfc6808>>.

Authors' Addresses

Guy Almes
Texas A&M

Email: almes@acm.org

Sunil Kalidindi
Ixia

Email: skalidindi@ixiacom.com

Matt Zekauskas
Internet2

Email: matt@internet2.edu

Al Morton (editor)
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
Obsoletes: 2680 (if approved)
Intended status: Standards Track
Expires: February 21, 2016

G. Almes
Texas A&M
S. Kalidindi
Ixia
M. Zekauskas
Internet2
A. Morton, Ed.
AT&T Labs
August 20, 2015

A One-Way Loss Metric for IPPM
draft-ietf-ippm-2680-bis-05

Abstract

This memo (RFC 2680 bis) defines a metric for one-way loss of packets across Internet paths. It builds on notions introduced and discussed in the IPPM Framework document, RFC 2330; the reader is assumed to be familiar with that document. This memo makes RFC 2680 obsolete.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 21, 2016.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
1.1. Motivation	3
1.2. General Issues Regarding Time	4
2. A Singleton Definition for One-way Packet Loss	6
2.1. Metric Name:	6
2.2. Metric Parameters:	6
2.3. Metric Units:	6
2.4. Definition:	6
2.5. Discussion:	6
2.6. Methodologies:	7
2.7. Errors and Uncertainties:	8
2.8. Reporting the metric:	9
2.8.1. Type-P	10
2.8.2. Loss Threshold	10
2.8.3. Calibration Results	10
2.8.4. Path	10
3. A Definition for Samples of One-way Packet Loss	11
3.1. Metric Name:	11
3.2. Metric Parameters:	11
3.3. Metric Units:	11
3.4. Definition:	12
3.5. Discussion:	12
3.6. Methodologies:	13
3.7. Errors and Uncertainties:	13
3.8. Reporting the metric:	14
4. Some Statistics Definitions for One-way Packet Loss	14
4.1. Type-P-One-way-Packet Loss-Ratio	14
5. Security Considerations	15
6. Acknowledgements	16
7. Changes from RFC 2680	16
8. IANA Considerations	18
9. References	18
9.1. Normative References	18
9.2. Informative References	19
Authors' Addresses	20

1. Introduction

This memo defines a metric for one-way packet loss across Internet paths. It builds on notions introduced and discussed in the IPPM

Framework document, [RFC2330]; the reader is assumed to be familiar with that document, and its recent update [RFC7312].

This memo is intended to be parallel in structure to a companion document for One-way Delay ("A One-way Delay Metric for IPPM") [RFC2679]; the reader is assumed to be familiar with that document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Although [RFC2119] was written with protocols in mind, the key words are used in this document for similar reasons. They are used to ensure the results of measurements from two different implementations are comparable, and to note instances when an implementation could perturb the network.

The structure of the memo is as follows:

- + A 'singleton' analytic metric, called Type-P-One-way-Packet-Loss, is introduced to measure a single observation of packet transmission or loss.

- + Using this singleton metric, a 'sample', called Type-P-One-way-Packet-Loss-Poisson-Stream, is introduced to measure a sequence of singleton transmissions and/or losses measured at times taken from a Poisson process.

- + Using this sample, several 'statistics' of the sample are defined and discussed.

This progression from singleton to sample to statistics, with clear separation among them, is important.

Whenever a technical term from the IPPM Framework document is first used in this memo, it will be tagged with a trailing asterisk. For example, "term*" indicates that "term" is defined in the Framework.

1.1. Motivation

Understanding one-way packet loss of Type-P* packets from a source host* to a destination host is useful for several reasons:

- + Some applications do not perform well (or at all) if end-to-end loss between hosts is large relative to some threshold value.

- + Excessive packet loss may make it difficult to support certain real-time applications (where the precise threshold of "excessive" depends on the application).

- + The larger the value of packet loss, the more difficult it is for transport-layer protocols to sustain high bandwidths.

- + The sensitivity of real-time applications and of transport-layer protocols to loss become especially important when very large delay-bandwidth products must be supported.

The measurement of one-way loss instead of round-trip loss is motivated by the following factors:

- + In today's Internet, the path from a source to a destination may be different than the path from the destination back to the source ("asymmetric paths"), such that different sequences of routers are used for the forward and reverse paths. Therefore round-trip measurements actually measure the performance of two distinct paths together. Measuring each path independently highlights the performance difference between the two paths which may traverse different Internet service providers, and even radically different types of networks (for example, research versus commodity networks, or networks with asymmetric link capacities, or wireless vs. wireline access).

- + Even when the two paths are symmetric, they may have radically different performance characteristics due to asymmetric queueing.

- + Performance of an application may depend mostly on the performance in one direction. For example, a TCP-based communication will experience reduced throughput if congestion occurs in one direction of its communication. Trouble shooting may be simplified if the congested direction of TCP transmission can be identified.

- + In quality-of-service (QoS) enabled networks, provisioning in one direction may be radically different than provisioning in the reverse direction, and thus the QoS guarantees differ. Measuring the paths independently allows the verification of both guarantees.

It is outside the scope of this document to say precisely how loss metrics would be applied to specific problems.

1.2. General Issues Regarding Time

{Comment: the terminology below differs from that defined by ITU-T documents (e.g., G.810, "Definitions and terminology for synchronization networks" and I.356, "B-ISDN ATM layer cell transfer performance"), but is consistent with the IPPM Framework document. In general, these differences derive from the different backgrounds; the ITU-T documents historically have a telephony origin, while the authors of this document (and the Framework) have a computer systems

background. Although the terms defined below have no direct equivalent in the ITU-T definitions, after our definitions we will provide a rough mapping. However, note one potential confusion: our definition of "clock" is the computer operating systems definition denoting a time-of-day clock, while the ITU-T definition of clock denotes a frequency reference.}

Whenever a time (i.e., a moment in history) is mentioned here, it is understood to be measured in seconds (and fractions) relative to UTC.

As described more fully in the Framework document, there are four distinct, but related notions of clock uncertainty:

synchronization*

measures the extent to which two clocks agree on what time it is. For example, the clock on one host might be 5.4 msec ahead of the clock on a second host. {Comment: A rough ITU-T equivalent is "time error".}

accuracy*

measures the extent to which a given clock agrees with UTC. For example, the clock on a host might be 27.1 msec behind UTC. {Comment: A rough ITU-T equivalent is "time error from UTC".}

resolution*

specification of the smallest unit by which the clock's time is updated. It gives a lower bound on the clock's uncertainty. For example, the clock on an old Unix host might tick only once every 10 msec, and thus have a resolution of only 10 msec. {Comment: A very rough ITU-T equivalent is "sampling period".}

skew*

measures the change of accuracy, or of synchronization, with time. For example, the clock on a given host might gain 1.3 msec per hour and thus be 27.1 msec behind UTC at one time and only 25.8 msec an hour later. In this case, we say that the clock of the given host has a skew of 1.3 msec per hour relative to UTC, which threatens accuracy. We might also speak of the skew of one clock relative to another clock, which threatens synchronization. {Comment: A rough ITU-T equivalent is "time drift".}

2. A Singleton Definition for One-way Packet Loss

2.1. Metric Name:

Type-P-One-way-Packet-Loss

2.2. Metric Parameters:

- + Src, the IP address of a host
- + Dst, the IP address of a host
- + T, a time
- + Tmax, a loss threshold waiting time

2.3. Metric Units:

The value of a Type-P-One-way-Packet-Loss is either a zero (signifying successful transmission of the packet) or a one (signifying loss).

2.4. Definition:

>>The *Type-P-One-way-Packet-Loss* from Src to Dst at T is 0<< means that Src sent the first bit of a Type-P packet to Dst at wire-time* T and that Dst received that packet.

>>The *Type-P-One-way-Packet-Loss* from Src to Dst at T is 1<< means that Src sent the first bit of a type-P packet to Dst at wire-time T and that Dst did not receive that packet (within the loss threshold waiting time, Tmax).

2.5. Discussion:

Thus, Type-P-One-way-Packet-Loss is 0 exactly when Type-P-One-way-Delay is a finite value, and it is 1 exactly when Type-P-One-way-Delay is undefined.

The following issues are likely to come up in practice:

- + A given methodology will have to include a way to distinguish between a packet loss and a very large (but finite) delay. As noted by Mahdavi and Paxson [RFC2678], simple upper bounds (such as the 255 seconds theoretical upper bound on the lifetimes of IP packets [RFC0791]) could be used, but good engineering, including an understanding of packet lifetimes, will be needed in practice. {Comment: Note that, for many applications of these metrics, there

may be no harm in treating a large delay as packet loss. An audio playback packet, for example, that arrives only after the playback point may as well have been lost. See section 4.1.1 of [RFC6703] for examination of unusual packet delays and application performance estimation.}

+ If the packet arrives, but is corrupted, then it is counted as lost. {Comment: one is tempted to count the packet as received since corruption and packet loss are related but distinct phenomena. If the IP header is corrupted, however, one cannot be sure about the source or destination IP addresses and is thus on shaky grounds about knowing that the corrupted received packet corresponds to a given sent test packet. Similarly, if other parts of the packet needed by the methodology to know that the corrupted received packet corresponds to a given sent test packet, then such a packet would have to be counted as lost. Counting these packets as lost but packet with corruption in other parts of the packet as not lost would be inconsistent.} Section 15 of [RFC2330] defines the "standard-formed" packet which is applicable to all metrics. Note: At this time, the definition of standard-formed packets only applies to IPv4, but also see [I-D.morton-ippm-2330-stdform-typep].

+ If the packet is duplicated along the path (or paths) so that multiple non-corrupt copies arrive at the destination, then the packet is counted as received.

+ If the packet is fragmented and if, for whatever reason, reassembly does not occur, then the packet will be deemed lost.

2.6. Methodologies:

As with other Type-P-* metrics, the detailed methodology will depend on the Type-P (e.g., protocol number, UDP/TCP port number, size, Differentiated Services (DS) Field [RFC2780])).

Generally, for a given Type-P, one possible methodology would proceed as follows:

+ Arrange that Src and Dst have clocks that are synchronized with each other. The degree of synchronization is a parameter of the methodology, and depends on the threshold used to determine loss (see below).

+ At the Src host, select Src and Dst IP addresses, and form a test packet of Type-P with these addresses.

+ At the Dst host, arrange to receive the packet.

- + At the Src host, place a timestamp in the prepared Type-P packet, and send it towards Dst (ideally minimizing time before sending).

- + If the packet arrives within a reasonable period of time, the one-way packet-loss is taken to be zero (and take a timestamp as soon as possible upon the receipt of the packet).

- + If the packet fails to arrive within a reasonable period of time, T_{max} , the one-way packet-loss is taken to be one. Note that the threshold of "reasonable" here is a parameter of the metric.

{Comment: The definition of reasonable is intentionally vague, and is intended to indicate a value "Th" so large that any value in the closed interval [Th-delta, Th+delta] is an equivalent threshold for loss. Here, delta encompasses all error in clock synchronization and timestamp acquisition and assignment along the measured path. If there is a single value, T_{max} , after which the packet must be counted as lost, then we reintroduce the need for a degree of clock synchronization similar to that needed for one-way delay, and virtually all practical measurement systems combine methods for delay and loss. Therefore, if a measure of packet loss parameterized by a specific non-huge "reasonable" time-out value is needed, one can always measure one-way delay and see what percentage of packets from a given stream exceed a given time-out value. This point is examined in detail in [RFC6703], including analysis preferences to assign undefined delay to packets that fail to arrive with the difficulties emerging from the informal "infinite delay" assignment, and an estimation of an upper bound on waiting time for packets in transit. Further, enforcing a specific constant waiting time on stored singletons of one-way delay is compliant with this specification and may allow the results to serve more than one reporting audience.}

Issues such as the packet format, the means by which Dst knows when to expect the test packet, and the means by which Src and Dst are synchronized are outside the scope of this document. {Comment: We plan to document elsewhere our own work in describing such more detailed implementation techniques and we encourage others to as well.}

2.7. Errors and Uncertainties:

The description of any specific measurement method should include an accounting and analysis of various sources of error or uncertainty. The Framework document provides general guidance on this point.

For loss, there are three sources of error:

- + Synchronization between clocks on Src and Dst.

- + The packet-loss threshold (which is related to the synchronization between clocks).

- + Resource limits in the network interface or software on the receiving instrument.

The first two sources are interrelated and could result in a test packet with finite delay being reported as lost. Type-P-One-way-Packet-Loss is 1 if the test packet does not arrive, or if it does arrive and the difference between Src timestamp and Dst timestamp is greater than the "reasonable period of time", or loss threshold. If the clocks are not sufficiently synchronized, the loss threshold may not be "reasonable" - the packet may take much less time to arrive than its Src timestamp indicates. Similarly, if the loss threshold is set too low, then many packets may be counted as lost. The loss threshold must be high enough, and the clocks synchronized well enough so that a packet that arrives is rarely counted as lost. (See the discussions in the previous two sections.)

Since the sensitivity of packet loss measurement alone to lack of clock synchronization is less than for delay, we refer the reader to the treatment of synchronization errors in the One-way Delay metric [RFC2330] for more details.

The last source of error, resource limits, cause the packet to be dropped by the measurement instrument, and counted as lost when in fact the network delivered the packet in reasonable time.

The measurement instruments should be calibrated such that the loss threshold is reasonable for application of the metrics and the clocks are synchronized enough so the loss threshold remains reasonable.

In addition, the instruments should be checked to ensure the that the possibility a packet arrives at the network interface, but is lost due to congestion on the interface or to other resource exhaustion (e.g., buffers) on the instrument is low.

2.8. Reporting the metric:

The calibration and context in which the metric is measured MUST be carefully considered, and SHOULD always be reported along with metric results. We now present four items to consider: Type-P of the test packets, the loss threshold, instrument calibration, and the path traversed by the test packets. This list is not exhaustive; any additional information that could be useful in interpreting applications of the metrics should also be reported (see [RFC6703] for extensive discussion of reporting considerations for different audiences).

2.8.1. Type-P

As noted in the Framework document, section 13 of [RFC2330], the value of the metric may depend on the type of IP packets used to make the measurement, or "Type-P". The value of Type-P-One-way-Delay could change if the protocol (UDP or TCP), port number, size, or arrangement for special treatment (e.g., IP DS Field [RFC2780], ECN [RFC3168], or RSVP) changes. Additional packet distinctions identified in future extensions of the Type-P definition will apply. The exact Type-P used to make the measurements MUST be accurately reported.

2.8.2. Loss Threshold

The threshold, T_{max} , (or methodology to distinguish) between a large finite delay and loss MUST be reported.

2.8.3. Calibration Results

The degree of synchronization between the Src and Dst clocks MUST be reported. If possible, possibility that a test packet that arrives at the Dst network interface is reported as lost due to resource exhaustion on Dst SHOULD be reported.

2.8.4. Path

Finally, the path traversed by the packet SHOULD be reported, if possible. In general it is impractical to know the precise path a given packet takes through the network. The precise path may be known for certain Type-P on short or stable paths. If Type-P includes the record route (or loose-source route) option in the IP header, and the path is short enough, and all routers* on the path support record (or loose-source) route, then the path will be precisely recorded. This is impractical because the route must be short enough, many routers do not support (or are not configured for) record route, and use of this feature would often artificially worsen the performance observed by removing the packet from common-case processing. However, partial information is still valuable context. For example, if a host can choose between two links* (and hence two separate routes from Src to Dst), then the initial link used is valuable context. {Comment: Backbone path selection services come and go. A historical example was Merit's NetNow setup, where a Src on one NAP can reach a Dst on another NAP by either of several different backbone networks.}

3. A Definition for Samples of One-way Packet Loss

Given the singleton metric `Type-P-One-way-Packet-Loss`, we now define one particular sample of such singletons. The idea of the sample is to select a particular binding of the parameters `Src`, `Dst`, and `Type-P`, then define a sample of values of parameter `T`. The means for defining the values of `T` is to select a beginning time `T0`, a final time `Tf`, and an average rate `lambda`, then define a pseudo-random Poisson process of rate `lambda`, whose values fall between `T0` and `Tf`. The time interval between successive values of `T` will then average $1/\text{lambda}$.

Note that Poisson sampling is only one way of defining a sample. Poisson has the advantage of limiting bias, but other methods of sampling will be appropriate for different situations. For example, a truncated Poisson distribution may be needed to avoid reactive network state changes during intervals of inactivity, see section 4.6 of [RFC7312]. Sometimes, the goal is sampling with a known bias, and [RFC3432] describes a method for periodic sampling with random start times.

3.1. Metric Name:

`Type-P-One-way-Packet-Loss-Poisson-Stream`

3.2. Metric Parameters:

- + `Src`, the IP address of a host
- + `Dst`, the IP address of a host
- + `T0`, a time
- + `Tf`, a time
- + `Tmax`, a loss threshold waiting time
- + `lambda`, a rate in reciprocal seconds

3.3. 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 monotonic increasing. Note that T would be a valid parameter to Type-P-One-way-Packet-Loss, and that L would be a valid value of Type-P-One-way-Packet-Loss.

3.4. Definition:

Given T_0 , T_f , and λ , we compute a pseudo-random Poisson process beginning at or before T_0 , with average arrival rate λ , and ending at or after T_f . Those time values greater than or equal to T_0 and less than or equal to T_f are then selected. At each of the times in this process, we obtain the value of Type-P-One-way-Packet-Loss at this time. The value of the sample is the sequence made up of the resulting $\langle \text{time}, \text{loss} \rangle$ pairs. If there are no such pairs, the sequence is of length zero and the sample is said to be empty.

3.5. Discussion:

The reader should be familiar with the in-depth discussion of Poisson sampling in the Framework document [RFC2330], which includes methods to compute and verify the pseudo-random Poisson process.

We specifically do not constrain the value of λ , except to note the extremes. If the rate is too large, then the measurement traffic will perturb the network, and itself cause congestion. If the rate is too small, then you might not capture interesting network behavior. {Comment: We expect to document our experiences with, and suggestions for, λ elsewhere, culminating in a "best current practices" document.}

Since a pseudo-random number sequence is employed, the sequence of times, and hence the value of the sample, is not fully specified. Pseudo-random number generators of good quality will be needed to achieve the desired qualities.

The sample is defined in terms of a Poisson process both to avoid the effects of self-synchronization and also capture a sample that is statistically as unbiased as possible. The Poisson process is used to schedule the loss measurements. The test packets will generally not arrive at Dst according to a Poisson distribution, since they are influenced by the network. Time-slotted links described in section 3.4 [RFC7312] can greatly modify the sample characteristics. The main concern is that un-biased packet streams with randomized inter-packet time intervals will be converted to some new distribution after encountering a time-slotted links, possibly with strong periodic characteristics instead.

{Comment: there is, of course, no claim that real Internet traffic arrives according to a Poisson arrival process.

It is important to note that, in contrast to this metric, loss ratios observed by transport connections do not reflect unbiased samples. For example, TCP transmissions both (1) occur in bursts, which can induce loss due to the burst volume that would not otherwise have been observed, and (2) adapt their transmission rate in an attempt to minimize the loss ratio observed by the connection.}

All the singleton Type-P-One-way-Packet-Loss metrics in the sequence will have the same values of Src, Dst, and Type-P.

Note also that, given one sample that runs from T_0 to T_f , and given new time values T_0' and T_f' such that $T_0 \leq T_0' \leq T_f' \leq T_f$, the subsequence of the given sample whose time values fall between T_0' and T_f' are also a valid Type-P-One-way-Packet-Loss-Poisson-Stream sample.

3.6. Methodologies:

The methodologies follow directly from:

- + the selection of specific times, using the specified Poisson arrival process, and
- + the methodologies discussion already given for the singleton Type-P-One-way-Packet-Loss metric.

Care must be given to correctly handle out-of-order arrival of test packets; it is possible that the Src could send one test packet at $TS[i]$, then send a second one (later) at $TS[i+1]$, while the Dst could receive the second test packet at $TR[i+1]$, and then receive the first one (later) at $TR[i]$. Metrics for reordering may be found in [RFC4737].

3.7. Errors and Uncertainties:

In addition to sources of errors and uncertainties associated with methods employed to measure the singleton values that make up the sample, care must be given to analyze the accuracy of the Poisson arrival process of the wire-times of the sending of the test packets. Problems with this process could be caused by several things, including problems with the pseudo-random number techniques used to generate the Poisson arrival process. The Framework document shows how to use the Anderson-Darling test to verify the accuracy of the Poisson process over small time frames. {Comment: The goal is to ensure that the test packets are sent "close enough" to a Poisson schedule, and avoid periodic behavior.}

3.8. Reporting the metric:

The calibration and context for the underlying singletons MUST be reported along with the stream. (See "Reporting the metric" for Type-P-One-way-Packet-Loss.)

4. Some Statistics Definitions for One-way Packet Loss

Given the sample metric Type-P-One-way-Packet-Loss-Poisson-Stream, we now offer several statistics of that sample. These statistics are offered mostly to be illustrative of what could be done. See [RFC6703] for additional discussion of statistics that are relevant to different audiences.

4.1. Type-P-One-way-Packet Loss-Ratio

Given a Type-P-One-way-Packet-Loss-Poisson-Stream, the average of all the L values in the Stream is the ratio of losses to total packets in the stream. In addition, the Type-P-One-way-Packet-Loss-Ratio is undefined if the sample is empty.

Example: suppose we take a sample and the results are:

```
Stream1 = <
<T1, 0>
<T2, 0>
<T3, 1>
<T4, 0>
<T5, 0>
>
```

Then the average of loss results would be 0.2, the loss ratio.

Note that, since healthy Internet paths should be operating at loss ratios below 1% (particularly if high delay-bandwidth products are to be sustained), the sample sizes needed might be larger than one would like. Thus, for example, if one wants to discriminate between various fractions of 1% over one-minute periods, then several hundred samples per minute might be needed. This would result in larger values of lambda than one would ordinarily want.

Note that although the loss threshold should be set such that any errors in loss are not significant, if the possibility that a packet which arrived is counted as lost due to resource exhaustion is significant compared to the loss ratio of interest, Type-P-One-way-Packet-Loss-Ratio will be meaningless.

5. Security Considerations

Conducting Internet measurements raises both security and privacy concerns. This memo does not specify an implementation of the metrics, so it does not directly affect the security of the Internet nor of applications which run on the Internet. However, implementations of these metrics must be mindful of security and privacy concerns.

There are two types of security concerns: potential harm caused by the measurements, and potential harm to the measurements. The measurements could cause harm because they are active, and inject packets into the network. The measurement parameters MUST be carefully selected so that the measurements inject trivial amounts of additional traffic into the networks they measure. If they inject "too much" traffic, they can skew the results of the measurement, and in extreme cases cause congestion and denial of service.

The measurements themselves could be harmed by routers giving measurement traffic a different priority than "normal" traffic, or by an attacker injecting artificial measurement traffic. If routers can recognize measurement traffic and treat it separately, the measurements will not reflect actual user traffic. If an attacker injects artificial traffic that is accepted as legitimate, the loss ratio will be artificially lowered. Therefore, the measurement methodologies SHOULD include appropriate techniques to reduce the probability measurement traffic can be distinguished from "normal" traffic. Authentication techniques, such as digital signatures, may be used where appropriate to guard against injected traffic attacks.

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

Collecting measurements or using measurement results for reconnaissance to assist in subsequent system attacks is quite

common. Access to measurement results, or control of the measurement systems to perform reconnaissance should be guarded against. See Section 7 of [I-D.ietf-lmap-framework] (security considerations of the LMAP Framework) for system requirements that help to avoid measurement system compromise.

6. Acknowledgements

For [RFC2680], thanks are due to Matt Mathis for encouraging this work and for calling attention on so many occasions to the significance of packet loss. Thanks are due also to Vern Paxson for his valuable comments on early drafts, and to Garry Couch and Will Leland for several useful suggestions.

For RFC 2680 bis, thanks to Joachim Fabini, Ruediger Geib, Nalini Elkins, and Barry Constantine for sharing their measurement experience as part of their careful reviews. Brian Carpenter and Scott Bradner provided useful feedback at IETF Last Call.

7. Changes from RFC 2680

Note: This section's placement currently preserves minimal differences between this memo and RFC 2680. The RFC Editor should place this section in an appropriate place.

The text above constitutes RFC 2680 bis proposed for advancement on the IETF Standards Track.

[RFC7290] provides the test plan and results supporting [RFC2680] advancement along the standards track, according to the process in [RFC6576]. The conclusions of [RFC7290] list four minor modifications for inclusion:

1. Section 6.2.3 of [RFC7290] asserts that the assumption of post-processing to enforce a constant waiting time threshold is compliant, and that the text of the RFC should be revised slightly to include this point. The applicability of post-processing was added in the last list item of section 2.6, above.
2. Section 6.5 of [RFC7290] indicates that Type-P-One-way-Packet-Loss-Average statistic is more commonly called Packet Loss Ratio, so it is re-named in RFC2680bis (this small discrepancy does not affect candidacy for advancement) The re-naming was implemented in section 4.1, above.
3. 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. This reference

was added in the last list item of section 2.6, in section 2.8, and in section 4 above.

4. There are currently two errata with status "Verified" and "Held for document update" for [RFC2680], and these minor revisions were incorporated in section 1 and section 2.7.

A number of updates to the [RFC2680] text have been implemented in the text, to reference key IPPM RFCs that were approved after [RFC2680] (see sections 3 and 3.6, above), and to address comments on the IPPM mailing list describing current conditions and experience.

1. Near the end of section 1.1, update of a network example using ATM and clarification of TCP's affect on queue occupation and importance of one-way delay measurement.
2. Clarification of the definition of "resolution" in section 1.2.
3. Explicit inclusion of the maximum waiting time input parameter in sections 2.2, 2.4, and 3.2, reflecting recognition of this parameter in more recent RFCs and ITU-T Recommendation Y.1540.
4. Addition of reference to RFC 6703 in the discussion of packet life time and application timeouts in section 2.5.
5. Replaced "precedence" with updated terminology (DS Field) in 2.6 and 2.8.1 (with reference).
6. Added parenthetical guidance on minimizing interval between timestamp placement to send time or reception time in section 2.6. Also, the text now recognizes the timestamp acquisition process and that practical systems measure both delay and loss (thus require the max waiting time parameter).
7. Added reference to RFC 3432 Periodic sampling alongside Poisson sampling in section 3, and also noting that a truncated Poisson distribution may be needed with modern networks as described in the IPPM Framework update, [RFC7312].
8. Recognition that Time-slotted links described in [RFC7312] can greatly modify the sample characteristics, in section 3.5.
9. Add reference to RFC 4737 Reordering metric in the related discussion of section 3.6, Methodologies.
10. Expanded and updated the material on Privacy, and added cautions on use of measurements for reconnaissance in section 5, Security Considerations.

Section 5.4.4 of [RFC6390] suggests a common template for performance metrics partially derived from previous IPPM and BMWG RFCs, but also contains some new items. All of the [RFC6390] Normative points are covered, but not quite in the same section names or orientation. Several of the Informative points are covered. Maintaining the familiar outline of IPPM literature has value and minimizes unnecessary differences between this revised RFC and current/future IPPM RFCs.

8. IANA Considerations

This memo makes no requests of IANA.

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<http://www.rfc-editor.org/info/rfc2330>>.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <<http://www.rfc-editor.org/info/rfc2678>>.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, DOI 10.17487/RFC2679, September 1999, <<http://www.rfc-editor.org/info/rfc2679>>.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, DOI 10.17487/RFC2680, September 1999, <<http://www.rfc-editor.org/info/rfc2680>>.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, DOI 10.17487/RFC2780, March 2000, <<http://www.rfc-editor.org/info/rfc2780>>.

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, DOI 10.17487/RFC3432, November 2002, <<http://www.rfc-editor.org/info/rfc3432>>.
- [RFC6576] Geib, R., Ed., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, DOI 10.17487/RFC6576, March 2012, <<http://www.rfc-editor.org/info/rfc6576>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<http://www.rfc-editor.org/info/rfc7312>>.

9.2. Informative References

- [I-D.ietf-lmap-framework]
Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for Large-Scale Measurement of Broadband Performance (LMAP)", draft-ietf-lmap-framework-14 (work in progress), April 2015.
- [I-D.morton-ippm-2330-stdform-typep]
Morton, A., Fabini, J., Elkins, N., Ackermann, M., and V. Hegde, "Updates for IPPM's Active Metric Framework: Packets of Type-P and Standard-Formed Packets", draft-morton-ippm-2330-stdform-typep-00 (work in progress), August 2015.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<http://www.rfc-editor.org/info/rfc4737>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<http://www.rfc-editor.org/info/rfc6390>>.

- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, DOI 10.17487/RFC6703, August 2012, <<http://www.rfc-editor.org/info/rfc6703>>.
- [RFC7290] Ciavattone, L., Geib, R., Morton, A., and M. Wieser, "Test Plan and Results for Advancing RFC 2680 on the Standards Track", RFC 7290, DOI 10.17487/RFC7290, July 2014, <<http://www.rfc-editor.org/info/rfc7290>>.

Authors' Addresses

Guy Almes
Texas A&M

Email: almes@acm.org

Sunil Kalidindi
Ixia

Email: skalidindi@ixiacom.com

Matt Zekauskas
Internet2

Email: matt@internet2.edu

Al Morton (editor)
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: August 9, 2015

A. Morton
AT&T Labs
February 5, 2015

Rate Measurement Test Protocol Problem Statement and Requirements
draft-ietf-ippm-rate-problem-10

Abstract

This memo presents an access rate-measurement problem statement for test protocols to measure IP Performance Metrics. Key rate measurement test protocol aspects include the ability to control packet characteristics on the tested path, such as asymmetric rate and asymmetric packet size.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 9, 2015.

Copyright Notice

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

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

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

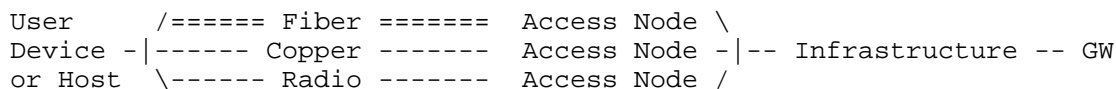
Table of Contents

1. Introduction 2
 2. Purpose and Scope 3
 3. Active Rate Measurement 5
 4. Measurement Method Categories 7
 5. Test Protocol Control & Generation Requirements 9
 6. Security Considerations 10
 7. Operational Considerations 11
 8. IANA Considerations 11
 9. Acknowledgements 12
 10. References 12
 10.1. Normative References 12
 10.2. Informative References 12
 Author's Address 13

1. Introduction

There are many possible rate measurement scenarios. This memo describes one rate measurement problem and presents a rate-measurement problem statement for test protocols to measure IP Performance Metrics (IPPM).

When selecting a form of access to the Internet, subscribers are interested in the performance characteristics of the various alternatives. Standardized measurements can be a basis for comparison between these alternatives. There is an underlying need to coordinate measurements that support such comparisons, and test control protocols to fulfill this need. The figure below depicts some typical measurement points of access networks.



The access-rate scenario or use case has received wide-spread attention of Internet access subscribers and seemingly all Internet industry players, including regulators. This problem is being approached with many different measurement methods. The eventual protocol solutions to this problem (and the systems that utilize the protocol) may not directly involve users, such as when tests reach

from the Infrastructure to a service-specific device, such as a residential gateway. However, no aspect of the problem precludes users from developing a test protocol controlled via command line interfaces on both ends. Thus, a very wide range of test protocols, active measurement methods and system solutions are the possible outcomes of this problem statement.

2. Purpose and Scope

The scope and purpose of this memo is to define the measurement problem statement for test protocols conducting access rate measurement on production networks. Relevant test protocols include [RFC4656] and [RFC5357], but the problem is stated in a general way so that it can be addressed by any existing test protocol, such as [RFC6812].

This memo discusses possibilities for methods of measurement, but does not specify exact methods which would normally be part of the solution, not the problem.

We are interested in access measurement scenarios with the following characteristics:

- o The Access portion of the network is the focus of this problem statement. The user typically subscribes to a service with bi-directional access partly described by rates in bits per second. The rates may be expressed as raw capacity or restricted capacity as described in [RFC6703]. These are the quantities that must be measured according to one or more standard metrics, and for which measurement methods must also be agreed as a part of the solution.
- o Referring to the reference path illustrated below and defined in [I-D.ietf-ippm-lmap-path], possible measurement points include a Subscriber's host, the access service demarcation point, Intra IP access where a globally routable address is present, or the gateway between the measured access network and other networks.

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

GRA = Globally Routable Address, GW = Gateway

- o Rates at some links near the edge of the provider's network can often be several orders of magnitude less than link rates in the aggregation and core portions of the network.
- o Asymmetrical access rates on ingress and egress are prevalent.

- o In many scenarios of interest, extremely large scale of access services requires low complexity devices participating at the user end of the path, and those devices place limits on clock and control timing accuracy.

This problem statement assumes that the most-likely bottleneck device or link is adjacent to the remote (user-end) measurement device, or is within one or two router/switch hops of the remote measurement device.

Other use cases for rate measurement involve situations where the packet switching and transport facilities are leased by one operator from another and the link capacity available cannot be directly determined (e.g., from device interface utilization). These scenarios could include mobile backhaul, Ethernet Service access networks, and/or extensions of layer 2 or layer 3 networks. The results of rate measurements in such cases could be employed to select alternate routing, investigate whether capacity meets some previous agreement, and/or adapt the rate of traffic sources if a capacity bottleneck is found via the rate measurement. In the case of aggregated leased networks, available capacity may also be asymmetric. In these cases, the tester is assumed to have a sender and receiver location under their control. We refer to this scenario below as the aggregated leased network case.

This memo describes protocol support for active measurement methods, consistent with the IPPM working group's traditional charter. Active measurements require synthetic traffic streams dedicated to testing, and do not make measurements on user traffic. See section 2 of [RFC2679], where the concept of a stream is first introduced in IPPM literature as the basis for collecting a sample (defined in section 11 of [RFC2330]).

As noted in [RFC2330] the focus of access traffic management may influence the rate measurement results for some forms of access, as it may differ between user and test traffic if the test traffic has different characteristics, primarily in terms of the packets themselves (see section 13 of [RFC2330] for the considerations on packet type, or Type-P).

There are several aspects of Type-P where user traffic may be examined and selected for special treatment that may affect transmission rates. Various aspects of Type-P are known to influence Equal-Cost Multi-Path (ECMP) routing with possible rate measurement variability across parallel paths. Without being exhaustive, the possibilities include:

- o Packet length

- o IP addresses
- o Transport protocol (e.g. where TCP packets may be routed differently from UDP)
- o Transport Protocol port numbers

This issue requires further discussion when specific solutions/methods of measurement are proposed, but for this problem statement it is sufficient to identify the problem and indicate that the solution may require an extremely close emulation of user traffic, in terms of one or more factors above.

Although the user may have multiple instances of network access available to them, the primary problem scope is to measure one form of access at a time. It is plausible that a solution for the single access problem will be applicable to simultaneous measurement of multiple access instances, but treatment of this scenario is beyond the current scope this document.

A key consideration is whether active measurements will be conducted with user traffic present (In-Service testing), or not present (Out-of-Service testing), such as during pre-service testing or maintenance that interrupts service temporarily. Out-of-Service testing includes activities described as "service commissioning", "service activation", and "planned maintenance". Opportunistic In-Service testing when there is no user traffic present (e.g., outside normal business hours) throughout the test interval is essentially equivalent to Out-of-Service testing. Both In-Service and Out-of-Service testing are within the scope of this problem.

It is a non-goal to solve the measurement protocol specification problem in this memo.

It is a non-goal to standardize methods of measurement in this memo. However, the problem statement mandates support for one category of rate measurement methods in the test protocol and adequate control features for the methods in the control protocol (assuming the control and test protocols are separate).

3. Active Rate Measurement

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

Coordination between source and destination devices through control messages and other basic capabilities described in the methods of

IPPM RFCs [RFC2679][RFC2680], and assumed for test protocols such as [RFC5357] and [RFC4656], are taken as given.

Most forms of active testing intrude on user performance to some degree, especially In-Service testing. One key tenet of IPPM methods is to minimize test traffic effects on user traffic in the production network. Section 5 of [RFC2680] lists the problems with high measurement traffic rates ("too much traffic"), and the most relevant for rate measurement is the tendency for measurement traffic to skew the results, followed by the possibility of introducing congestion on the access link. Section 4 of [RFC3148] provides additional considerations. The user of protocols for In-Service testing MUST respect these traffic constraints. Obviously, categories of rate measurement methods that use less active test traffic than others with similar accuracy are preferred for In-Service testing, and the specifications of this memo encourage traffic reduction through asymmetric control capabilities.

Out-of-Service tests where the test path shares no links with In-Service user traffic, have none of the congestion or skew concerns. Both types should address practical matters common to all test efforts, such as conducting measurements within a reasonable time from the tester's point of view, and ensuring that timestamp accuracy is consistent with the precision needed for measurement [RFC2330]. Out-of-Service tests where some part of the test path is shared with In-Service traffic MUST respect the In-Service constraints described above.

The intended metrics to be measured have strong influence over the categories of measurement methods required. For example, using the terminology of [RFC5136], it may be possible to measure a Path Capacity Metric while In-Service if the level of background (user) traffic can be assessed and included in the reported result.

The measurement *architecture* MAY be either of one-way (e.g., [RFC4656]) or two-way (e.g., [RFC5357]), but the scale and complexity aspects of end-user or aggregated access measurement clearly favor two-way (with low-complexity user-end device and round-trip results collection, as found in [RFC5357]). However, the asymmetric rates of many access services mean that the measurement system MUST be able to evaluate performance in each direction of transmission. In the two-way architecture, both end devices MUST include the ability to launch test streams and collect the results of measurements in both (one-way) directions of transmission (this requirement is consistent with previous protocol specifications, and it is not a unique problem for rate measurements).

The following paragraphs describe features for the roles of test packet SENDER, RECEIVER, and results REPORTER.

SENDER:

Generate streams of test packets with various characteristics as desired (see Section 4). The SENDER MAY be located at the user end of the access path or elsewhere in the production network, such as at one end of an aggregated leased network segment.

RECEIVER:

Collect streams of test packets with various characteristics (as described above), and make the measurements necessary to support rate measurement at the receiving end of an access or aggregated leased network segment.

REPORTER:

Use information from test packets and local processes to measure delivered packet rates, and prepare results in the required format (the REPORTER role may be combined with another role, most likely the SENDER).

4. Measurement Method Categories

A protocol that addresses the rate measurement problem MUST serve the test stream generation and measurement functions (SENDER and RECEIVER). The follow-up phase of analyzing the measurement results to produce a report is outside the scope of this problem and memo (REPORTER).

For the purposes of this problem statement, we categorize the many possibilities for rate measurement stream generation as follows;

1. Packet pairs, with fixed intra-pair packet spacing and fixed or random time intervals between pairs in a test stream.
2. Multiple streams of packet pairs, with a range of intra-pair spacing and inter-pair intervals.
3. One or more packet ensembles in a test stream, using a fixed ensemble size in packets and one or more fixed intra-ensemble packet spacings (including zero spacing, meaning that back-to-back burst ensembles and constant rate ensembles fall in this category).

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

The test protocol SHALL support test packet ensemble generation (category 3), as this appears to minimize the demands on measurement accuracy. Other stream generation categories are OPTIONAL.

For all supported categories, the following is a list of additional variables that the protocol(s) MUST be able to specify, control, and generate:

- a. Variable payload lengths among packet streams
- b. Variable length (in packets) among packet streams or ensembles
- c. Variable IP header markings among packet streams
- d. Choice of UDP transport and variable port numbers, OR, choice of TCP transport and variable port numbers for two-way architectures only, OR BOTH. See below for additional requirements on TCP transport generation.
- e. Variable number of packet-pairs, ensembles, or streams used in a test session.

The ability to revise these variables during an established test session is OPTIONAL, as multiple test sessions could serve the same purpose. Another OPTIONAL feature is the ability to generate streams with VLAN tags and other markings.

For measurement systems employing TCP as the transport protocol, the ability to generate specific stream characteristics requires a sender with the ability to establish and prime the connection such that the desired stream characteristics are allowed. See Mathis' work in progress for more background [I-D.ietf-ippm-model-based-metrics].

Beyond simple connection handshake and options establishment, an "open-loop" TCP sender requires the SENDER ability to:

- o generate TCP packets with well-formed headers (all fields valid), including Acknowledgement aspects.
- o produce packet streams at controlled rates and variable inter-packet spacings, including packet ensembles (back-to-back at server rate).

- o continue the configured sending stream characteristics despite all control indications except receive window exhaust.

The corresponding TCP RECEIVER performs normally, having some ability to configure the receive window sufficiently large so as to allow the SENDER to transmit at will (up to a configured target).

It may also be useful (for diagnostic purposes) to provide a control for Bulk Transfer Capacity measurement with fully-specified (and congestion-controlled) TCP senders and receivers, as envisioned in [RFC3148], but this would be a brute-force assessment which does not follow the conservative tenets of IPPM measurement [RFC2330].

Measurements for each UDP test packet transferred between SENDER and RECEIVER MUST be compliant with the singleton measurement methods described in IPPM RFCs [RFC2679][RFC2680]. The time-stamp information or loss/arrival status for each packet MUST be available for communication to the REPORTER function.

5. Test Protocol Control & Generation Requirements

In summary, the test protocol must support the measurement features described in the sections above. This requires:

1. Communicating all test variables to the SENDER and RECEIVER
2. Results collection in a one-way architecture
3. Remote device control for both one-way and two-way architectures
4. Asymmetric packet rates in a two-way measurement architecture, or coordinated one-way test capabilities with the same effect (asymmetric rates may be achieved through directional control of packet rate or packet size)

The ability to control and generate asymmetric rates in a two-way architecture is REQUIRED. Two-way architectures are RECOMMENDED to include control and generation capability for both asymmetric and symmetric packet sizes, because packet size often matters in the scope of this problem and test systems SHOULD be equipped to detect directional size dependency through comparative measurements.

Asymmetric packet size control is indicated when the result of a measurement may depend on the size of the packets used in each direction, i.e. when any of the following conditions hold:

- o there is a link in the path with asymmetrical capacity in opposite directions (in combination with one or more of the conditions

below, but their presence or specific details may be unknown to the tester),

- o there is a link in the path which aggregates (or divides) packets into link-level frames, and may have a capacity that depends on packet size, rate, or timing,
- o there is a link in the path where transmission in one direction influences performance in the opposite direction,
- o there is a device in the path where transmission capacity depends on packet header processing capacity (in other words, the capacity is sensitive to packet size),
- o the target application stream is nominally MTU size packets in one direction vs. ACK stream in the other, (noting that there are a vanishing number of symmetrical-rate application streams for which rate measurement is wanted or interesting, but such streams might have some relevance at this time),
- o the distribution of packet losses is critical to rate assessment, and possibly other circumstances revealed by measurements comparing streams with symmetrical size and asymmetrical size.

Implementations may support control and generation for only symmetric packet sizes when none of the above conditions hold.

The test protocol SHOULD enable measurement of the [RFC5136] Capacity metric, either Out-of-Service, In-Service, or both. Other [RFC5136] metrics are OPTIONAL.

6. Security Considerations

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

Privacy considerations for measurement systems, particularly when Internet users participate in the tests in some way, are described in [I-D.ietf-lmap-framework].

There may be a serious issue if a proprietary Service Level Agreement involved with the access network segment provider were somehow leaked in the process of rate measurement. To address this, test protocols SHOULD NOT convey this information in a way that could be discovered by unauthorized parties.

7. Operational Considerations

All forms of testing originate traffic on the network, through their communications for control and results collection, or from dedicated measurement packet streams, or both. Testing traffic primarily falls in one of two categories, subscriber traffic or network management traffic. There is an on-going need to engineer networks so that various forms of traffic are adequately served, and publication of this memo does not change this need. Service subscribers and authorized users SHOULD obtain their network operator's or service provider's permission before conducting tests. Likewise, a service provider or third party SHOULD obtain the subscriber's permission to conduct tests, since they might temporarily reduce service quality. The protocol SHOULD communicate the permission status once the overall system has obtained it, either explicitly or through other means.

Subscribers, their service providers and network operators, and sometimes third parties, all seek to measure network performance. Capacity testing with active traffic often affects the packet transfer performance of streams traversing shared components of the test path, to some degree. The degradation can be minimized by scheduling such tests infrequently, and restricting the amount of measurement traffic required to assess capacity metrics. As a result, occasional short-duration estimates with minimal traffic are preferred to measurements based on frequent file transfers of many Megabytes with similar accuracy. New measurement methodologies intended for standardization should be evaluated individually for potential operational issues. However, the scheduled frequency of testing is as important as the methods used (and schedules are not typically submitted for standardization).

The new test protocol feature of asymmetrical packet size generation in two-way testing is recommended in this memo. It can appreciably reduce the load and packet processing demands of each test and therefore reduce the likelihood of degradation in one direction of the tested path. Current IETF standardized test protocols (e.g., [RFC5357], also [RFC6812]) do not possess the asymmetric size generation capability with two-way testing.

8. IANA Considerations

This memo makes no requests of IANA.

9. Acknowledgements

Dave McDysan provided comments and text for the aggregated leased use case. Yaakov Stein suggested many considerations to address, including the In-Service vs. Out-of-Service distinction and its implication on test traffic limits and protocols. Bill Cerveny, Marcelo Bagnulo, Kostas Pentikousis (a persistent reviewer), and Joachim Fabini have contributed insightful, clarifying comments that made this a better draft. Barry Constantine also provided suggestions for clarification.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

10.2. Informative References

- [I-D.ietf-ippm-lmap-path] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for Large-Scale Measurement of Broadband Performance", draft-ietf-ippm-lmap-path-07 (work in progress), October 2014.

[I-D.ietf-ippm-model-based-metrics]

Mathis, M. and A. Morton, "Model Based Bulk Performance Metrics", draft-ietf-ippm-model-based-metrics-03 (work in progress), July 2014.

[I-D.ietf-lmap-framework]

Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-10 (work in progress), January 2015.

[RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, February 2008.

[RFC6812] Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare, S., and E. Yedavalli, "Cisco Service-Level Assurance Protocol", RFC 6812, January 2013.

Author's Address

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

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

A. Morton
AT&T Labs
M. Bagnulo
UC3M
P. Eardley
BT
March 9, 2015

Initial Performance Metric Registry Entries
draft-mornuley-ippm-initial-registry-01

Abstract

This memo defines the Initial Entries for the Performance Metrics Registry.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	5
2. Scope	6
3. Registry Categories and Columns	6
4. UDP Round-trip Latency Registry Entry	7
4.1. Summary	7
4.1.1. ID (Identifier)	7
4.1.2. Name	7
4.1.3. URI	7
4.1.4. Description	8
4.2. Metric Definition	8
4.2.1. Reference Definition	8
4.2.2. Fixed Parameters	8
4.3. Method of Measurement	9
4.3.1. Reference Method	9
4.3.2. Packet Generation Stream	10
4.3.3. Traffic Filtering (observation) Details	10
4.3.4. Sampling Distribution	10
4.3.5. Run-time Parameters and Data Format	10
4.3.6. Roles	11
4.4. Output	11
4.4.1. Type/Value (two diff terms used)	11
4.4.2. Data Format	12
4.4.3. Reference	13
4.4.4. Metric Units	13
4.5. Administrative items	13
4.5.1. Status	13
4.5.2. Requestor (keep?)	13
4.5.3. Revision	13
4.5.4. Revision Date	13
4.6. Comments and Remarks	13
5. Packet Delay Variation Registry Entry	13
5.1. Summary	14
5.1.1. ID (Identifier)	14
5.1.2. Name	14
5.1.3. URI	14
5.1.4. Description	14
5.2. Metric Definition	14
5.2.1. Reference Definition	14
5.2.2. Fixed Parameters	15

5.3.	Method of Measurement	15
5.3.1.	Reference Method	15
5.3.2.	Packet Generation Stream	15
5.3.3.	Traffic Filtering (observation) Details	16
5.3.4.	Sampling Distribution	16
5.3.5.	Run-time Parameters and Data Format	16
5.3.6.	Roles	16
5.4.	Output	17
5.4.1.	Type/Value (two diff terms used)	17
5.4.2.	Data Format	17
5.4.3.	Reference	18
5.4.4.	Metric Units	18
5.5.	Administrative items	18
5.5.1.	Status	18
5.5.2.	Requestor (keep?)	19
5.5.3.	Revision	19
5.5.4.	Revision Date	19
5.6.	Comments and Remarks	19
6.	DNS Response Latency Registry Entry	19
6.1.	Summary	19
6.1.1.	ID (Identifier)	19
6.1.2.	Name	19
6.1.3.	URI	20
6.1.4.	Description	20
6.2.	Metric Definition	20
6.2.1.	Reference Definition	20
6.2.2.	Fixed Parameters	21
6.3.	Method of Measurement	22
6.3.1.	Reference Method	22
6.3.2.	Packet Generation Stream	23
6.3.3.	Traffic Filtering (observation) Details	23
6.3.4.	Sampling Distribution	23
6.3.5.	Run-time Parameters and Data Format	24
6.3.6.	Roles	25
6.4.	Output	25
6.4.1.	Type/Value (two diff terms used)	25
6.4.2.	Data Format	25
6.4.3.	Reference	26
6.4.4.	Metric Units	26
6.5.	Administrative items	26
6.5.1.	Status	26
6.5.2.	Requestor (keep?)	27
6.5.3.	Revision	27
6.5.4.	Revision Date	27
6.6.	Comments and Remarks	27
7.	partly BLANK Registry Entry	27
7.1.	Summary	27
7.1.1.	ID (Identifier)	27

- 7.1.2. Name 27
- 7.1.3. URI 27
- 7.1.4. Description 27
- 7.2. Metric Definition 28
 - 7.2.1. Reference Definition 28
 - 7.2.2. Fixed Parameters 28
- 7.3. Method of Measurement 29
 - 7.3.1. Reference Method 29
 - 7.3.2. Packet Generation Stream 29
 - 7.3.3. Traffic Filtering (observation) Details 29
 - 7.3.4. Sampling Distribution 29
 - 7.3.5. Run-time Parameters and Data Format 30
 - 7.3.6. Roles 30
- 7.4. Output 30
 - 7.4.1. Type/Value (two diff terms used) 30
 - 7.4.2. Data Format 30
 - 7.4.3. Reference 30
 - 7.4.4. Metric Units 30
- 7.5. Administrative items 31
 - 7.5.1. Status 31
 - 7.5.2. Requestor (keep?) 31
 - 7.5.3. Revision 31
 - 7.5.4. Revision Date 31
- 7.6. Comments and Remarks 31
- 8. BLANK Registry Entry 31
 - 8.1. Summary 31
 - 8.1.1. ID (Identifier) 31
 - 8.1.2. Name 31
 - 8.1.3. URI 31
 - 8.1.4. Description 32
 - 8.2. Metric Definition 32
 - 8.2.1. Reference Definition 32
 - 8.2.2. Fixed Parameters 32
 - 8.3. Method of Measurement 32
 - 8.3.1. Reference Method 32
 - 8.3.2. Packet Generation Stream 32
 - 8.3.3. Traffic Filtering (observation) Details 32
 - 8.3.4. Sampling Distribution 32
 - 8.3.5. Run-time Parameters and Data Format 33
 - 8.3.6. Roles 33
 - 8.4. Output 33
 - 8.4.1. Type/Value (two diff terms used) 33
 - 8.4.2. Data Format 33
 - 8.4.3. Reference 33
 - 8.4.4. Metric Units 33
 - 8.5. Administrative items 33
 - 8.5.1. Status 33
 - 8.5.2. Requestor (keep?) 33

8.5.3.	Revision	33
8.5.4.	Revision Date	34
8.6.	Comments and Remarks	34
9.	Example RTCP-XR Registry Entry	34
9.1.	Registry Indexes	34
9.1.1.	Identifier	34
9.1.2.	Name	34
9.1.3.	URI	34
9.1.4.	Status	34
9.1.5.	Requestor	34
9.1.6.	Revision	34
9.1.7.	Revision Date	35
9.1.8.	Description	35
9.1.9.	Reference Specification(s)	35
9.2.	Metric Definition	35
9.2.1.	Reference Definition	35
9.2.2.	Fixed Parameters	35
9.3.	Method of Measurement	36
9.3.1.	Reference Method	36
9.3.2.	Stream Type and Stream Parameters	36
9.3.3.	Output Type and Data Format	36
9.3.4.	Metric Units	37
9.3.5.	Run-time Parameters and Data Format	37
9.4.	Comments and Remarks	38
10.	Security Considerations	39
11.	IANA Considerations	39
12.	Acknowledgements	40
13.	References	40
13.1.	Normative References	40
13.2.	Informative References	41
	Authors' Addresses	42

1. Introduction

Note: Efforts to synchronize structure and terminology with [I-D.ietf-ippm-metric-registry] will likely be incomplete until both drafts are stable.

This memo defines the Initial set of entries for the Performance Metric Registry. The registry will contain Active Performance Metrics, especially those defined in RFCs prepared in the IP Performance Metrics (IPPM) Working Group of the IETF, according to their framework [RFC2330]. Three aspects make IPPM metric registration difficult: (1) Use of the Type-P notion to allow users to specify their own packet types. (2) Use of Flexible input variables, called Parameters in IPPM definitions, some which determine the quantity measured and others which should not be specified until execution of the measurement. (3) Allowing

flexibility in choice of statistics to summarize the results on a stream of measurement packets. This memo uses terms and definitions from the IPPM literature, primarily [RFC2330], and the reader is assumed familiar with them or may refer questions there as necessary.

Although there are several standard templates for organizing specifications of performance metrics (see [RFC2679] for an example of the traditional IPPM template, based to large extent on the Benchmarking Methodology Working Group's traditional template in [RFC1242], and see [RFC6390] for a similar template), none of these templates were intended to become the basis for the columns of an IETF-wide registry of metrics. As we examined the aspects of metric specifications which need to be registered, it was clear that none of the existing metric templates fully satisfies the particular needs of a registry.

2. Scope

[I-D.ietf-ippm-metric-registry] defines the overall structure for a Performance Metric Registry and provides guidance for the process to examine proposed metrics and maintain Registered Metrics.

This document defines the initial set of Performance Metrics Registry entries; all are active metrics, or those where the packets measured have been specially generated for the purpose.

A row in the registry corresponds to one Registered Performance Metric, with entries in the various columns specifying the metric.

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

3. Registry Categories and Columns

This section defines the categories and columns of the registry. Below, categories are described at the 3.x heading level, and columns are at the 3.x.y heading level. The Figure below illustrates this organization. An entry (row) therefore gives a complete description of a Registered Metric.

Each column serves as a check-list item and helps to avoid omissions during registration and expert review. In some cases an entry (row) may have some columns without specific entries, marked Not Applicable (NA).

THIS NEEDS UPDATING

Registry Categories and Columns, shown as

Category	
-----	-----
Column	Column

Comments and Remarks

4. UDP Round-trip Latency Registry Entry

This section gives an initial registry entry for the UDP Round-trip Latency.

Note: If each Registry entry should only produce a "raw" output or a statistical summary, then the "Output" Category can be split and this section can become two closely-related metrics.

4.1. Summary

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

4.1.1. ID (Identifier)

<insert numeric identifier, an integer>

4.1.2. Name

<insert name according to metric naming convention>

Act_IP_UDP_Round-trip_Delay_Raw_95th-percentile_Poisson

URL: ??

4.1.3. URI

URI: Prefix urn:iETF:params:performance:metric...<name>

4.1.4. Description

This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the Round-trip delay for all successfully exchanged packets and the 95th percentile of their conditional delay distribution.

4.2. Metric Definition

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

4.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

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

[RFC2681]

<specific section reference and additional clarifications, if needed>

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

4.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

Type-P:

- o IPv4 header values:

- * DSCP: set to 0

- * TTL set to 255
- * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Checksum: the checksum must be calculated
- o Payload
 - * Sequence number: 8-byte integer
 - * Timestamp: 8 byte integer. Expressed as 64-bit NTP timestamp as per section 6 of RFC 5905 [RFC5905]
 - * No padding (total of 9 bytes)

Timeout, Tmax: 3 seconds

4.3. Method of Measurement

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

4.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Timeout defined under Fixed Parameters.

The method requires sequence numbers or other send-order information to be retained at the Src or included with each packet to disambiguate packet reordering if it occurs. Sequence number is part of the payload described under Fixed Parameters.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which shall be included in the method of measurement for this metric.

4.3.2. Packet Generation Stream

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

<list of generation parameters and section/spec references if needed>

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. the reciprocal of lambda is the average packet rate, thus the Run-time Parameter is 1/lambda.

>>> Check with Sam, most likely it is this...

Method 3 is used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Run-time Parameters), and the Src sends each packet at the computed times.

4.3.3. Traffic Filtering (observation) Details

The measured results based on a filtered version of the packets observed, and this section provides the filter details (when present).

<section reference>.

NA

4.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

NA

4.3.5. Run-time Parameters and Data Format

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

<list of run-time parameters, and their data formats>

- o Src, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)

- o Dst, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o T0, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]). When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval.
- o Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]), interpreted as the Duration of the measurement interval.
- o 1/lambda, average packet rate (for Poisson Streams). (1/lambda = 1 packet per second, if fixed)
- o Upper limit on Poisson distribution (values above this limit will be clipped and set to the limit value). (if fixed, Upper limit = 30 seconds.)

The format for 1/lambda and Upper limit of Poisson Dist. are the short format in [RFC5905] (32 bits) and is as follows: the first 16 bits represent the integer number of seconds; the next 16 bits represent the fractional part of a second.

>>> should Poisson run-time params be fixed instead? probably yes if modeling a specific version of MBA tests.

4.3.6. Roles

<lists the names of the different roles from the measurement method>

Src - launches each packet and waits for return transmissions from Dst.

Dst - waits for each packet from Src and sends a return packet to Src.

4.4. Output

This category specifies all details of the Output of measurements using the metric.

4.4.1. Type/Value (two diff terms used)

<insert name of the output type, raw or a selected summary statistic>

Raw -- for each packet sent, pairs of values.

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value corresponding to the 95th percentile.

4.4.2. Data Format

<describe the data format for each type of result>

For all outputs ---

- o T₀, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o T_f, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])

Raw -- for each packet sent, pairs of values as follows:

- o T, the time when the packet was sent from Src, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o dT, a value of Round-trip delay, format is *similar to* the 32-bit short NTP Time format in Section 6 of [RFC5905] and is as follows: the first 16 bits represent the *signed* integer number of seconds; the next 16 bits represent the fractional part of a second.
- o dT is undefined when the packet is not received at Src in waiting time T_{mxax} seconds (need undefined code)

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95.

Data format is a 32-bit signed value, *similar to* the 32-bit short NTP Time format in Section 6 of [RFC5905] and is as follows: the first 16 bits represent the *signed* integer number of seconds; the next 16 bits represent the fractional part of a second.

4.4.3. Reference

<pointer to section/spec where output type/format is defined>

See the Data Format column for references.

4.4.4. Metric Units

<insert units for the measured results, and the reference specification>.

Round-trip Delay, dT, is expressed in seconds.

The 95th Percentile of Round-trip Delay is expressed in seconds.

4.5. Administrative items

4.5.1. Status

<current or deprecated>

4.5.2. Requestor (keep?)

name or RFC, etc.

4.5.3. Revision

1.0

4.5.4. Revision Date

YYYY-MM-DD

4.6. Comments and Remarks

Additional (Informational) details for this entry

5. Packet Delay Variation Registry Entry

This section gives an initial registry entry for a Packet Delay Variation metric.

Note: If each Registry entry should only produce a "raw" output or a statistical summary, then the "Output" Category can be split and this section can become two closely-related metrics.

5.1. Summary

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

<skipping some Summary columns for now>

5.1.1. ID (Identifier)

<insert numeric identifier, an integer>

5.1.2. Name

<insert name according to metric naming convention>

Act_IP-UDP-One-way-pdv-95th-percentile-Poisson

URL: ??

5.1.3. URI

URI: Prefix urn:ietf:params:performance:metric<add name>

5.1.4. Description

An assessment of packet delay variation with respect to the minimum delay observed on the stream.

5.2. Metric Definition

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

5.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998. [RFC2330]

Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002. [RFC3393]

Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009. [RFC5481]

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

<specific section reference and additional clarifications, if needed>

See sections 2.4 and 3.4 of [RFC3393]. Singleton delay differences measured are referred to by the variable name "ddT".

5.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

- o F, a selection function defining unambiguously the packets from the stream selected for the metric. See section 4.2 of [RFC5481] for the PDV form.
- o L, a packet length in bits. L = 200 bits.
- o Tmax, a maximum waiting time for packets to arrive at Dst, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost). Tmax = 3 seconds.
- o Type-P, as defined in [RFC2330], which includes any field that may affect a packet's treatment as it traverses the network. The packets are IP/UDP, with DSCP = 0 (BE).

5.3. Method of Measurement

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

5.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

See section 2.6 and 3.6 of [RFC3393] for singleton elements.

5.3.2. Packet Generation Stream

<list of generation parameters and section/spec references if needed>

Poisson distributed as described in [RFC2330], with the following Parameters.

- o λ , a rate in reciprocal seconds (for Poisson Streams).
 $\lambda = 1$ packet per second
- o Upper limit on Poisson distribution (values above this limit will be clipped and set to the limit value). Upper limit = 30 seconds.

5.3.3. Traffic Filtering (observation) Details

<insert the measured results based on a filtered version of the packets observed, and this section provides the filter details (when present), and section reference>.

NA

5.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

NA

5.3.5. Run-time Parameters and Data Format

<list of run-time parameters, and any reference(s)>.

- o Src, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o Dst, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o T, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]). When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval.
- o Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]), interpreted as the Duration of the measurement interval.

5.3.6. Roles

<lists the names of the different roles from the measurement method>

Src - the host that sends the stream of packets.

Dst - the host that receives the stream of packets.

5.4. Output

This category specifies all details of the Output of measurements using the metric.

5.4.1. Type/Value (two diff terms used)

<insert name of the output type, raw or a selected summary statistic>

Raw -- for each packet sent, pairs of values.

Percentile -- for the conditional distribution of all packets with a valid value of one-way delay (undefined delays are excluded), a single value corresponding to the 95th percentile of the singletons, ddT.

5.4.2. Data Format

<describe the data format for each type of result>

For all Output types

- o T, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])

Raw -

- o T1, the wire time of the first packet in a pair, measured at MP(Src) as it leaves for Dst (64-bit NTP Timestamp Format, see section 6 of [RFC5905]).
- o T2, the wire time of the second packet in a pair, measured at MP(Src) as it leaves for Dst (64-bit NTP Timestamp Format, see section 6 of [RFC5905]).
- o I(i),I(i+1), $i \geq 0$, pairs of times which mark the beginning and ending of the intervals in which the packet stream from which the measurement is taken occurs. Here, $I(0) = T0$ and assuming that n is the largest index, $I(n) = Tf$ (pairs of 64-bit NTP Timestamp Format, see section 6 of [RFC5905]).
- o When the one-way delay of a packet in the calculation pair for ddT is undefined, then ddT is undefined for that pair.

Percentile -- for the conditional distribution of all packets with a valid value of one-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where pdv should be substituted for "ipdv").

The percentile = 95.

Data format is a 32-bit signed floating point value, *similar to* the 32-bit short NTP Time format in Section 6 of [RFC5905] and is as follows: the first 16 bits represent the *signed* integer number of seconds; the next 16 bits represent the fractional part of a second.

5.4.3. Reference

<pointer to section/spec where output type/format is defined>

see Data Format column.

5.4.4. Metric Units

<insert units for the measured results, and the reference specification>.

See section 3.3 of [RFC3393] for singleton elements, ddT. The units are seconds, and the same units are used for 95th percentile.

[RFC2330] recommends that when a time is given, it will be expressed in UTC.

The timestamp format (for T, Tf, etc.) is the same as in [RFC5905] (64 bits) and is as follows: the first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the next 32 bits represent the fractional part of a second that has elapsed since then.

5.5. Administrative items

5.5.1. Status

<current or deprecated>

5.5.2. Requestor (keep?)

<name of individual or RFC, etc.>

5.5.3. Revision

1.0

5.5.4. Revision Date

YYYY-MM-DD

5.6. Comments and Remarks

<Additional (Informational) details for this entry>

Lost packets represent a challenge for delay variation metrics. See section 4.1 of [RFC3393] and the delay variation applicability statement[RFC5481] for extensive analysis and comparison of PDV and an alternate metric, IPDV.

6. DNS Response Latency Registry Entry

This section gives an initial registry entry for DNS Response Latency. RFC 2681 [RFC2681] defines a Round-trip delay metric. We build on that metric by specifying several of the input parameters to precisely define a metric for measuring DNS latency.

6.1. Summary

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

<skipping some admin columns for now>

6.1.1. ID (Identifier)

<insert numeric identifier, an integer>

6.1.2. Name

<insert name according to metric naming convention>

URL: ??

6.1.3. URI

URI: Prefix urn:ietf:params:performance:metric

6.1.4. Description

This metric assesses the response time, the interval from the query transmission to the response.

6.2. Metric Definition

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

6.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987. (and updates)

[RFC1035]

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

[RFC2681]

<specific section reference and additional clarifications, if needed>

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

For DNS Response Latency, the entities in [RFC1035] must be mapped to [RFC2681]. The Local Host with its User Program and Resolver take the role of "Src", and the Foreign Name Server takes the role of "Dst".

Note that although the definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

6.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

Type-P:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Source port: 53
 - * Destination port: 53
 - * Checksum: the checksum must be calculated
- o Payload: The payload contains a DNS message as defined in RFC 1035 [RFC1035] with the following values:
 - * The DNS header section contains:
 - + QR: set to 0 (Query)
 - + OPCODE: set to 0 (standard query)
 - + AA: not set
 - + TC: not set
 - + RD: set to one (recursion desired)
 - + RA: not set
 - + RCODE: not set
 - + QDCOUNT: set to one (only one entry)
 - + ANCOUNT: not set
 - + NSCOUNT: not set

- + ARCOUNT: not set
- * The Question section contains:
 - + QNAME: the FQDN provided as input for the test
 - + QTYPE: the query type provided as input for the test
 - + QCLASS: set to IN
- * The other sections do not contain any Resource Records.

Observation: reply packets will contain a DNS response and may contain RRs.

Timeout: Tmax = 5 seconds (to help disambiguate queries)

6.3. Method of Measurement

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

6.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Timeout defined under Fixed Parameters.

The method requires sequence numbers or other send-order information to be retained at the Src or included with each packet to disambiguate packet reordering if it occurs. Sequence number is part of the payload described under Fixed Parameters.

DNS Messages bearing Queries provide for random ID Numbers, so more than one query may be launched while a previous request is outstanding when the ID Number is used.

IF a DNS response does not arrive within Tmax, the result is undefined. The Message ID SHALL be used to disambiguate the successive queries.

>>> This would require support of ID generation and population in the Message. An alternative would be to use a random Source port on the Query Message, but we would choose ONE before proceeding.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which shall be included in the method of measurement for this metric.

6.3.2. Packet Generation Stream

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

<list of generation parameters and section/spec references if needed>

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. the reciprocal of lambda is the average packet rate, thus the Run-time Parameter is 1/lambda.

>>> Check with Sam, most likely it is this...

Method 3 is used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Run-time Parameters), and the Src sends each packet at the computed times.

6.3.3. Traffic Filtering (observation) Details

The measured results based on a filtered version of the packets observed, and this section provides the filter details (when present).

<section reference>.

NA

6.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

NA

6.3.5. Run-time Parameters and Data Format

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

<list of run-time parameters, and their data formats>

- o Src, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o Dst, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)
- o T0, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]). When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval.
- o Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]), interpreted as the Duration of the measurement interval.
- o 1/lambda, average packet rate (for Poisson Streams). (1/lambda = 0.1 packet per second, if fixed)
- o Upper limit on Poisson distribution (values above this limit will be clipped and set to the limit value). (if fixed, Upper limit = 300 seconds.)
- o ID, the 16-bit identifier assigned by the program that generates the query, and which must vary in successive queries, see Section 4.1.1 of [RFC1035]. This identifier is copied into the corresponding reply and can be used by the requester to match-up replies to outstanding queries.

The format for 1/lambda and Upper limit of Poisson Dist. are the short format in [RFC5905] (32 bits) and is as follows: the first 16 bits represent the integer number of seconds; the next 16 bits represent the fractional part of a second.

>>> should Poisson run-time params be fixed instead? probably yes if modeling a specific version of MBA tests.

6.3.6. Roles

<lists the names of the different roles from the measurement method>

Src - launches each packet and waits for return transmissions from Dst.

Dst - waits for each packet from Src and sends a return packet to Src.

6.4. Output

This category specifies all details of the Output of measurements using the metric.

6.4.1. Type/Value (two diff terms used)

<insert name of the output type, raw or a selected summary statistic>

For all output types:

- o T0, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905])

Raw -- for each packet sent, pairs of values.

>>> and the status of the response, only assigning values to successful query-response pairs.

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value corresponding to the 95th percentile.

6.4.2. Data Format

<describe the data format for each type of result>

Raw -- for each packet sent, pairs of values as follows:

- o T, the time when the packet was sent from Src, 128-bit NTP Date Format, see section 6 of [RFC5905])
- o dT, a value of Round-trip delay, format is *similar to* the 32-bit short NTP Time format in Section 6 of [RFC5905] and is as follows: the first 16 bits represent the *signed* integer number of

seconds; the next 16 bits represent the fractional part of a second.

- o dT is undefined when the packet is not received at Src in waiting time Tmxax seconds (need undefined code for no-response or unsuccessful response)

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95.

Data format is a 32-bit signed floating point value, *similar to* the 32-bit short NTP Time format in Section 6 of [RFC5905] and is as follows: the first 16 bits represent the *signed* integer number of seconds; the next 16 bits represent the fractional part of a second.

6.4.3. Reference

<pointer to section/spec where output type/format is defined>

See the Data Format column for references.

6.4.4. Metric Units

<insert units for the measured results, and the reference specification>.

Round-trip Delay, dT, is expressed in seconds.

The 95th Percentile of Round-trip Delay is expressed in seconds.

6.5. Administrative items

6.5.1. Status

<current or deprecated>

6.5.2. Requestor (keep?)

name or RFC, etc.

6.5.3. Revision

1.0

6.5.4. Revision Date

YYYY-MM-DD

6.6. Comments and Remarks

Additional (Informational) details for this entry

7. partly BLANK Registry Entry

This section gives an initial registry entry for

7.1. Summary

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

<skipping the admin columns for now>

7.1.1. ID (Identifier)

<insert numeric identifier, an integer>

7.1.2. Name

<insert name according to metric naming convention>

URL: ??

7.1.3. URI

URI: Prefix urn:ietf:params:performance:metric

7.1.4. Description

TBD.

7.2. Metric Definition

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

7.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

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

<specific section reference and additional clarifications, if needed>

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

<<< Check how the Methodology also makes this clear (or not) >>>

7.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

Type-P:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:

- * Checksum: the checksum must be calculated
- o Payload
 - * Sequence number: 8-byte integer
 - * Timestamp: 8 byte integer. Expressed as 64-bit NTP timestamp as per section 6 of RFC 5905 [RFC5905]
 - * No padding (total of 9 bytes)

Timeout: 3 seconds

7.3. Method of Measurement

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

7.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

7.3.2. Packet Generation Stream

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

<list of generation parameters and section/spec references if needed>

7.3.3. Traffic Filtering (observation) Details

The measured results based on a filtered version of the packets observed, and this section provides the filter details (when present).

<section reference>.

7.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

7.3.5. Run-time Parameters and Data Format

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

<list of run-time parameters>

<reference(s)>.

7.3.6. Roles

<lists the names of the different roles from the measurement method>

7.4. Output

This category specifies all details of the Output of measurements using the metric.

7.4.1. Type/Value (two diff terms used)

<insert name of the output type, raw or a selected summary statistic>

7.4.2. Data Format

<describe the data format for each type of result>

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

7.4.3. Reference

<pointer to section/spec where output type/format is defined>

7.4.4. Metric Units

<insert units for the measured results, and the reference specification>.

7.5. Administrative items

7.5.1. Status

<current or deprecated>

7.5.2. Requestor (keep?)

name or RFC, etc.

7.5.3. Revision

1.0

7.5.4. Revision Date

YYYY-MM-DD

7.6. Comments and Remarks

Additional (Informational) details for this entry

8. BLANK Registry Entry

This section gives an initial registry entry for

8.1. Summary

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

<skipping the Summary columns for now>

8.1.1. ID (Identifier)

<insert numeric identifier, an integer>

8.1.2. Name

<insert name according to metric naming convention>

URL: ??

8.1.3. URI

URI: Prefix urn:iETF:params:performance:metric

8.1.4. Description

TBD.

8.2. Metric Definition

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

8.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

<specific section reference and additional clarifications, if needed>

8.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

8.3. Method of Measurement

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

8.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

8.3.2. Packet Generation Stream

<list of generation parameters and section/spec references if needed>

8.3.3. Traffic Filtering (observation) Details

<insert the measured results based on a filtered version of the packets observed, and this section provides the filter details (when present), and section reference>.

8.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

8.3.5. Run-time Parameters and Data Format

<list of run-time parameters, and any reference(s)>.

8.3.6. Roles

<lists the names of the different roles from the measurement method>

8.4. Output

This category specifies all details of the Output of measurements using the metric.

8.4.1. Type/Value (two diff terms used)

<insert name of the output type, raw or a selected summary statistic>

8.4.2. Data Format

<describe the data format for each type of result>

8.4.3. Reference

<pointer to section/spec where output type/format is defined>

8.4.4. Metric Units

<insert units for the measured results, and the reference specification>.

8.5. Administrative items

8.5.1. Status

<current or deprecated>

8.5.2. Requestor (keep?)

<name of individual or RFC, etc.>

8.5.3. Revision

1.0

8.5.4. Revision Date

YYYY-MM-DD

8.6. Comments and Remarks

Additional (Informational) details for this entry

9. Example RTCP-XR Registry Entry

This section is MAY BE DELETED or adapted before submission.

This section gives an example registry entry for the end-point metric described in RFC 7003 [RFC7003], for RTCP-XR Burst/Gap Discard Metric reporting.

9.1. Registry Indexes

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

9.1.1. Identifier

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

9.1.2. Name

A metric naming convention is TBD.

9.1.3. URI

Prefix urn:ietf:params:performance:metric

9.1.4. Status

current

9.1.5. Requestor

Alcelip Mornuley

9.1.6. Revision

1.0

9.1.7. Revision Date

2014-07-04

9.1.8. Description

TBD.

9.1.9. Reference Specification(s)

[RFC3611][RFC4566][RFC6776][RFC6792][RFC7003]

9.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters. Section 3.2 of [RFC7003] provides the reference information for this category.

9.2.1. Reference Definition

Packets Discarded in Bursts:

The total number of packets discarded during discard bursts. The measured value is unsigned value. If the measured value exceeds 0xFFFFFD, the value 0xFFFFFE MUST be reported to indicate an over-range measurement. If the measurement is unavailable, the value 0xFFFFF MUST be reported.

9.2.2. Fixed Parameters

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

Threshold: 8 bits, set to value = 3 packets.

The Threshold is equivalent to Gmin in [RFC3611], i.e., the number of successive packets that must not be discarded prior to and following a discard packet in order for this discarded packet to be regarded as part of a gap. Note that the Threshold is set in accordance with the Gmin calculation defined in Section 4.7.2 of [RFC3611].

Interval Metric flag: 2 bits, set to value 11=Cumulative Duration

This field is used to indicate whether the burst/gap discard metrics are Sampled, Interval, or Cumulative metrics [RFC6792]:

I=10: Interval Duration - the reported value applies to the most recent measurement interval duration between successive metrics reports.

I=11: Cumulative Duration - the reported value applies to the accumulation period characteristic of cumulative measurements.

Senders MUST NOT use the values I=00 or I=01.

9.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations. For the Burst/Gap Discard Metric, it appears that the only guidance on methods of measurement is in Section 3.0 of [RFC7003] and its supporting references. Relevant information is repeated below, although there appears to be no section titled "Method of Measurement" in [RFC7003].

9.3.1. Reference Method

Metrics in this block report on burst/gap discard in the stream arriving at the RTP system. Measurements of these metrics are made at the receiving end of the RTP stream. Instances of this metrics block use the synchronization source (SSRC) to refer to the separate auxiliary Measurement Information Block [RFC6776], which describes measurement periods in use (see [RFC6776], Section 4.2).

This metrics block relies on the measurement period in the Measurement Information Block indicating the span of the report. Senders MUST send this block in the same compound RTCP packet as the Measurement Information Block. Receivers MUST verify that the measurement period is received in the same compound RTCP packet as this metrics block. If not, this metrics block MUST be discarded.

9.3.2. Stream Type and Stream Parameters

Since RTCP-XR Measurements are conducted on live RTP traffic, the complete description of the stream is contained in SDP messages that proceed the establishment of a compatible stream between two or more communicating hosts. See Run-time Parameters, below.

9.3.3. Output Type and Data Format

The output type defines the type of result that the metric produces.

- o Value: Packets Discarded in Bursts

- o Data Format: 24 bits
- o Reference: Section 3.2 of [RFC7003]

9.3.4. Metric Units

The measured results are apparently expressed in packets, although there is no section of [RFC7003] titled "Metric Units".

9.3.5. Run-time Parameters and Data Format

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

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

SSRC of Source: 32 bits As defined in Section 4.1 of [RFC3611].

SDP Parameters: As defined in [RFC4566]

Session description v= (protocol version number, currently only 0)

o= (originator and session identifier : username, id, version number, network address)

s= (session name : mandatory with at least one UTF-8-encoded character)

i=* (session title or short information) u=* (URI of description)

e=* (zero or more email address with optional name of contacts)

p=* (zero or more phone number with optional name of contacts)

c=* (connection information--not required if included in all media)

b=* (zero or more bandwidth information lines) One or more Time descriptions ("t=" and "r=" lines; see below)

z=* (time zone adjustments)

k=* (encryption key)

a=* (zero or more session attribute lines)

Zero or more Media descriptions (each one starting by an "m=" line;
see below)

m= (media name and transport address)

i=* (media title or information field)

c=* (connection information -- optional if included at session level)

b=* (zero or more bandwidth information lines)

k=* (encryption key)

a=* (zero or more media attribute lines -- overriding the Session
attribute lines)

An example Run-time SDP description follows:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 192.0.2.5
s=SDP Seminar i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf e=j.doe@example.com (Jane
Doe)
c=IN IP4 233.252.0.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

9.4. Comments and Remarks

TBD.

10. Security Considerations

These registry entries represent no known security implications for Internet Security. Each referenced Metric contains a Security Considerations section.

11. IANA Considerations

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

Common Registry Indexes and Info

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

Metric Definition

- o Reference Definition
- o Fixed Parameters

Method of Measurement

- o Reference Method
- o Stream Type and Parameters
- o Output type and Data format
- o Metric Units

- o Run-time Parameters

- Comments and Remarks

12. Acknowledgements

The authors thank Brian Trammell for suggesting the term "Run-time Parameters", which led to the distinction between run-time and fixed parameters implemented in this memo, for raising the IPFIX metric with Flow Key as an example, and for many other productive suggestions. Thanks to Peter Koch, who provided several useful suggestions for disambiguating successive DNS Queries in the DNS Response time metric.

13. References

13.1. Normative References

- [I-D.ietf-ippm-metric-registry]
Bagnulo, M., Claise, B., Eardley, P., and A. Morton,
"Registry for Performance Metrics", Internet Draft (work
in progress) draft-ietf-ippm-metric-registry, 2014.
- [RFC1035] Mockapetris, P., "Domain names - implementation and
specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis,
"Framework for IP Performance Metrics", RFC 2330, May
1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way
Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way
Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip
Delay Metric for IPPM", RFC 2681, September 1999.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation
Metric for IP Performance Metrics (IPPM)", RFC 3393,
November 2002.

- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, August 2012.

13.2. Informative References

- [Brow00] Brownlee, N., "Packet Matching for NeTraMet Distributions", March 2000.
- [I-D.ietf-lmap-framework] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for Large-Scale Measurement of Broadband Performance (LMAP)", draft-ietf-lmap-framework-11 (work in progress), February 2015.
- [RFC1242] Bradner, S., "Benchmarking terminology for network interconnection devices", RFC 1242, July 1991.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC 5472, March 2009.

- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.
- [RFC6776] Clark, A. and Q. Wu, "Measurement Identity and Information Reporting Using a Source Description (SDES) Item and an RTCP Extended Report (XR) Block", RFC 6776, October 2012.
- [RFC6792] Wu, Q., Hunt, G., and P. Arden, "Guidelines for Use of the RTP Monitoring Framework", RFC 6792, November 2012.
- [RFC7003] Clark, A., Huang, R., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Discard Metric Reporting", RFC 7003, September 2013.

Authors' Addresses

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

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

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

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: <http://www.it.uc3m.es>

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: philip.eardley@bt.com