

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2017

B. M Gaonkar
S. Jacob
Juniper
G. Fioccola
Telecom Italia
Q. Wu
Huawei
P. Ananthasankaran
Nokia
June 24, 2017

Performance Measurement Models
draft-bhaprasud-ippm-pm-03

Abstract

This document defines the performance measurement models for service level packets on the network which can be implemented in different kind of network scenarios. Based on the performance matrix, the analytics data can be pulled from a live network which is not possible at present. This can be used for self evolving networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 26, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. Traffic Management Architecture	5
3.1. Selection Process	5
3.2. Metering Process	6
4. Performance Measurement Models	6
4.1. Complete data measurement (Monitoring all the traffic)	6
4.2. Color based data measurement	7
4.3. CoS based Data measurement	7
4.4. CoS and Color based Data measurement	8
5. Active and Passive performance measurements	8
6. Use Cases	8
7. Acknowledgements	9
8. Security Considerations	9
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Authors' Addresses	10

1. Introduction

Today performance monitoring or tracking of the performance experienced by customer traffic is a key technology to strengthen service offering and verify service level agreement between customers and service providers, perform troubleshooting. The lack of adequate monitoring tools to detect an interesting subset of a packet stream, as identified by a particular packet attribute (e.g., commit rate or DSCP) and measure that packet loss drives an effort to design a new method for the performance monitoring of live traffic, possibly easy to implement and deploy. The draft aims to provide fine granularity loss, delay and delay variation measurement and define a performance measurement model on customer traffic based on a set of constraints that are associated with service level agreement such as cos attribute, color attribute. Each customer traffic is corresponding to an interesting subset of the same packet stream. The customer or a interesting packet stream can be identified by a list of source or destination prefixes, or by ingress or egress interfaces, combing with packet attributes such as DSCP or commit rate). Unlike Color and COS identification specified in MEF 23.1, this draft doesn't define

new Color and CoS identification mechanism, instead, it stick to color definition in [RFC2697] and [RFC2698] and COS definition in [RFC2474].

The network would be provisioned with multiple services(e.g., real time service, interactive service) having different network performance criteria(e.g., bandwidth constraint or packet loss constraint for the end to end path) based on the customers' requirement. This models aims at performing Loss, Delay and delay variation measurement for these services (belonging to the same customer)independently for each defined network performance criteria.

The class-of-service and packet color classification defined in the network is a key factor to classify network traffic and drive traffic management mechanism to achieve corresponding network performance criteria for each service. This draft uses the class-of-service model and color based model for any given network to define the performance measurement for various services with the different network performance criteria requirements.

The proposed models is suitable mainly for passive performance measurements but can be considered for active and hybrid performance measurements as well.

This solution models loss, delay an delay variation measurement in different kinds of network scenarios. The different models explained here will help to analyse performance pattern, analyze the network congestion in a better way and model the network in a better way. For instance, Loss measurement is carried out between 2 end points. The underlying technology could be an active loss measurement or a passive loss measurement.

Any loss measurement will require 2 counters:

- o Number of packets transmitted from one end point.
- o Number of packets received at the other end point.

This draft explains the different ways to model the above data and get meaningful result for the loss, delay and delay variation measurement. The underlying technology could be an MPLS performance measurement, or an IP based performance measurement.

2. Conventions used in this document

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

Observation Point An Observation Point is a location in the network where data packets can be observed. Examples include a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router.

Persistence Data Store The persistence Data store is a scalable data store which collects time based data such as streaming data or time series data for network analytics.

Time Series Data Time Series Data is a sequence of data points with time stamps. The data points are limited to loss, delay and delay variation measurement results in this document.

Packet Stream A Packet Stream denotes a set of packets from the Observed Packet Stream that flows past some specified point within the Metering Process. An example of a Packet Stream is the output of the Selection Process.

Packet Content The Packet Content denotes the union of the packet header (which includes link layer, network layer, and other encapsulation headers) and the packet payload.

Color Identifier: It is used to identify the color that applies to the data packet. Color identifier can be assigned to service level packet based on commit rate and excess rate set for the traffic. For example, the service level packet will be set with "green" color if it is less than committed" rate; the Service Level packet will be set with "yellow" color if it is exceeding the "committed" rate but less than the "excess" rate. The service frame will be set with "red" color if it is exceeding both the "committed" and "excess" rates.

CoS Identifier: It is used to identify the CoS that applies to the data packet. CoS identifier can be assigned based on dot1p value in C-tag, or DSCP in IP header.

Complete data measurement: Complete data measurement is a data measurement method which monitors every packet and condense a large amount of information about packet arrivals into a small number of statistics. The aim of "monitoring every packet" is to ensure that the information reported is not dependent on the application.

Color based data measurement: Color based data measurement is a data measurement method which monitors the data packet with the same color identifier. Color identifier could be "green" color, "yellow" color and "red" color.

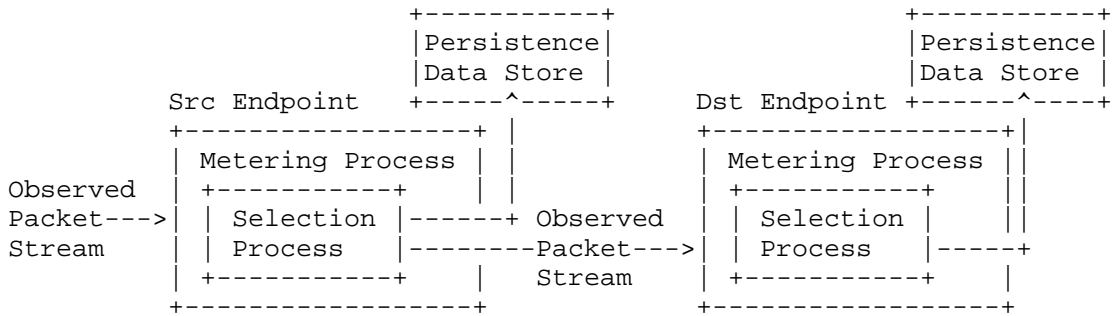
CoS based data measurement: Color based data measurement is a data measurement method which monitors the data packet with the same CoS identifier. COS identifier could be C-Tag Priority Code Point(PCP) or DSCP.

CoS and Color based Data measurement: CoS and Color based Data measurement is a data measurement method which monitors the data packet with the specific CoS Identifier and Specific Color Identifier as constraints. The measurement results with CoS Identifier and Color Identifier constraints constitute a Network Performance matrix.

3. Traffic Management Architecture

A stream of packets is observed at an Observation Point of the source endpoint and destination endpoints. Two observation points can also be placed at the same endpoint for node monitoring [I-D.ietf-ippm-alt-mark], i.e., one is at ingress interface of the endpoint and the other is at the egress interface of the endpoint. A Selection Process inspects each packet to determine whether or not it is to be selected for data analytics. The Selection Process is part of the Metering Process, which constructs a report stream on selected packets as output, using the Packet Content, and possibly other information such as the arrival timestamp. The report stream on selected packets will be stored in the persistence data store for real time data analysis or time sequence data analysis.

The following figure indicates the sequence of the three processes (Selection, Metering, and Storing).



3.1. Selection Process

This section defines the Selection Process and related objects.

Selection Process: A Selection Process takes the Observed Packet Stream as its input and selects a subset of that stream as its output.

Selection State: A Selection Process may maintain state information for use by the Selection Process. At a given time, the Selection State may depend on packets observed at and before that time, and other variables. Examples include sequence numbers of packets at the input of Selectors, a timestamp of observation of the packet at the Observation Point, indicators of whether the packet was selected by a given Selector.

Selector: A Selector defines the action of a Selection Process on a single packet of its input. If selected, the packet becomes an element of the output Packet Stream.

The Selector can make use of the following information in determining whether a packet is selected:

- * COS Identifier in the Packet Content;
- * Traffic attribute such as Color identifier;
- * Combination of CoS Identifier and Color Identifier

3.2. Metering Process

A Metering Process selects packets from the Observed Packet Stream using a Selection Process, and produces as output a Report Stream concerning the selected packets.

4. Performance Measurement Models

4.1. Complete data measurement (Monitoring all the traffic)

This model uses the complete data traffic between the 2 end-points to compute loss measurement, delay and delay variation. This will result in computation of loss, delay and delay variation measurement for the entire traffic in the network in one direction. This is primarily used in cases of backbone traffic where traffic from different services are aggregated and send into the core network. This will count all the packet, this gives the overall measurement between one endpoint to other.

4.2. Color based data measurement

This is same as the above section of "complete data measurement" with a minor difference, only monitoring the data packet with specific color identifier.

In this model the packets are counted in the following Way: Count specific data traffic with different color identifier between 2 end points for loss, delay and delay variation measurement. One example of Color based data measurement is to count two type of color based traffic:

- o Count all committed traffic between the 2 end-point for loss measurement.
- o Count all Excess traffic which is beyond the committed traffic for the specific network.
- o The probe carries the time stamps, which can later be used for calculating the service outage.
- o This method can be used for mapping the overall customer traffic along with EIR, based on the EIR provider can increase the bandwidth and charge him.

When both of these are combined then it becomes the model for complete traffic as mentioned in the above section.

In practice the Color of traffic can use any mechanism based on the network encapsulation. As long as the packets could be treated differently based on the underlying encapsulation this mechanism could be used.

This can be used for measuring the whole traffic of the customer who dont want cos level measurement. Ideally this can be used for provider who extend bandwidth for small providers, point to point services etc.

4.3. CoS based Data measurement

This model uses the data traffic in the network which is flowing in a specific CoS to measure the loss, delay and delay variation in the network. Based on the class of traffic in the network the transmitted and received packets are counted to calculate the packets transfered per service level. The time stamp will be captured along with the packet count to measure the service down time. This model measures the performance per service level. This data can be stored on the routers which can be used to plot the live analytics.

Primary use of this kind of measurement is to measure packet loss delay and delay variation for a specific service which needs to meet network performance requirements. The service could be a point-to-point layer2 service, an MPLS based service.

4.4. CoS and Color based Data measurement

This model uses a combination of both Color based data measurement and CoS based data measurement. Packets are counted for a specific CoS with a specific Color. This can count both in profile packet which are green and yellow which are out profile packets. This will not count the red packet which doesn't meet network performance requirements. The packets will be counted per service level with CIR and EIR along with time stamps to find the service outage and loss. The per service level counting for CoS and color will give more granular level data for plotting service graph and if some service is continuously exceeding the bandwidth this data can be used for charging the end customer for extra bandwidth usage or increase the bandwidth based on usage basis.

5. Active and Passive performance measurements

This model reinforces the use of well known methodologies for passive performance measurements. A very simple, flexible and straightforward mechanism is presented in [I-D.ietf-ippm-alt-mark]. The basic idea is to virtually split traffic flows into consecutive batches of packets: each block represents a measurable entity unambiguously recognizable thanks to the alternate marking. This approach, called Alternate Marking method, is efficient both for passive performance monitoring and for active performance monitoring. Most of the applications requires passive packet loss measurement for a better accuracy. Instead, in some cases, it is desirable to have only active delay measurements (e.g TWAMP or OWAMP), because it is enough.

6. Use Cases

Consider a provider running point to point service between router A and B for his customer "X". Customer "X" has voice traffic which requires special treatment, then he requires attention for database traffic. The customer "X" has SLA with the provider. Now the challenge faced by the provider is how to measure the traffic of customer "X" for each class and calculate the bandwidth, moreover the provider has to see whether the "X" is sending traffic which is exceeding the level so that he can make tariff accordingly. This problem is solved by the above models which can measure the packet for each class of traffic and tabulates the data. Later point of time this data can be pulled for evaluation.

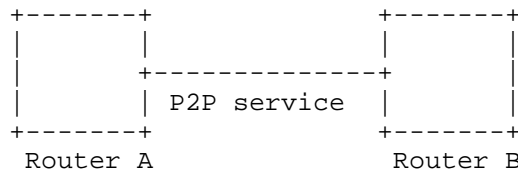


Figure 1: P2P

The same considerations can be applicable in a multipoint to multipoint scenario (e.g. VPN or Data Center interconnections). In this case Customer "X" has multiple ingress endpoints and multiple egress endpoints. The proposed matrix model is composed by the number of flows of "X" in the multipoint scenario and by class-of-service and color classification. So the SLA matrix is a reference for the analysis and evaluation phase.

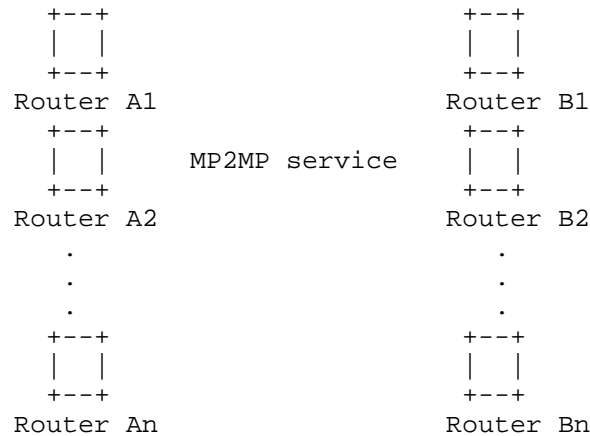


Figure 2: MP2MP

7. Acknowledgements

We would like to thank Brian Trammell for giving us the opportunity to present our draft. We would like to thank Greg Mirsky for the comments.

8. Security Considerations

This document does not introduce security issues beyond those discussed in [I-D.ietf-ippm-alt-mark].

9. References

9.1. Normative References

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

9.2. Informative References

[I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L.,
Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
"Alternate Marking method for passive performance
monitoring", draft-ietf-ippm-alt-mark-04 (work in
progress), March 2017.

Authors' Addresses

Bharat M Gaonkar
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
USA

Email: gbharat@juniper.net

Sudhin Jacob
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
USA

Email: gbharat@juniper.net

Giuseppe Fioccola
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Praveen Ananthasankaran
Nokia
Manyata Embassy Tech Park, Silver Oak (Wing A),
Outer Ring Road, Nagawara
Bangalore 560045
India

Email: praveen.ananthasankaran@nokia.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: June 10, 2018

G. Fioccola, Ed.
A. Capello
M. Cociglio
L. Castaldelli
Telecom Italia
M. Chen
L. Zheng
Huawei Technologies
G. Mirsky
ZTE
T. Mizrahi
Marvell
December 7, 2017

Alternate Marking method for passive and hybrid performance monitoring
draft-ietf-ippm-alt-mark-14

Abstract

This document describes a method to perform packet loss, delay and jitter measurements on live traffic. This method is based on Alternate Marking (Coloring) technique. A report is provided in order to explain an example and show the method applicability. This technology can be applied in various situations as detailed in this document and could be considered passive or hybrid depending on the application.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 June 10, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Overview of the method	5
3. Detailed description of the method	6
3.1. Packet loss measurement	6
3.1.1. Coloring the packets	11
3.1.2. Counting the packets	11
3.1.3. Collecting data and calculating packet loss	12
3.2. Timing aspects	13
3.3. One-way delay measurement	14
3.3.1. Single marking methodology	14
3.3.2. Double marking methodology	16
3.4. Delay variation measurement	17
4. Considerations	18
4.1. Synchronization	18
4.2. Data Correlation	19
4.3. Packet Re-ordering	20
5. Applications, implementation and deployment	20
5.1. Report on the operational experiment	21
5.1.1. Metric transparency	23
6. Hybrid measurement	24
7. Compliance with RFC6390 guidelines	24
8. Security Considerations	26
9. IANA Considerations	28
10. Acknowledgements	28
11. References	28
11.1. Normative References	28

11.2. Informative References	29
Authors' Addresses	31

1. Introduction

Nowadays, most Service Providers' networks carry traffic with contents that are highly sensitive to packet loss [RFC7680], delay [RFC7679], and jitter [RFC3393].

In view of this scenario, Service Providers need methodologies and tools to monitor and measure network performances with an adequate accuracy, in order to constantly control the quality of experience perceived by their customers. On the other hand, performance monitoring provides useful information for improving network management (e.g. isolation of network problems, troubleshooting, etc.).

A lot of work related to OAM, that includes also performance monitoring techniques, has been done by Standards Developing Organizations(SDOs): [RFC7276] provides a good overview of existing OAM mechanisms defined in IETF, ITU-T and IEEE. Considering IETF, a lot of work has been done on fault detection and connectivity verification, while a minor effort has been dedicated so far to performance monitoring. The IPPM WG has defined standard metrics to measure network performance; however, the methods developed in this WG mainly refer to focus on active measurement techniques. More recently, the MPLS WG has defined mechanisms for measuring packet loss, one-way and two-way delay, and delay variation in MPLS networks[RFC6374], but their applicability to passive measurements has some limitations, especially for pure connection-less networks.

The lack of adequate tools to measure packet loss with the desired accuracy drove an effort to design a new method for the performance monitoring of live traffic, easy to implement and deploy. The effort led to the method described in this document: basically, it is a passive performance monitoring technique, potentially applicable to any kind of packet based traffic, including Ethernet, IP, and MPLS, both unicast and multicast. The method addresses primarily packet loss measurement, but it can be easily extended to one-way delay and delay variation measurements as well.

The method has been explicitly designed for passive measurements but it can also be used with active probes. Passive measurements are usually more easily understood by customers and provide a much better accuracy, especially for packet loss measurements.

RFC 7799 [RFC7799] defines passive and hybrid methods of measurement. In particular, Passive Methods of Measurement are based solely on

observations of an undisturbed and unmodified packet stream of interest; Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods.

Taking into consideration these definitions, Alternate Marking Method could be considered Hybrid or Passive depending on the case. In case the marking method is obtained by changing existing field values of the packets (e.g. DSCP field), the technique is Hybrid. In case the marking field is dedicated, reserved and is included in the protocol specification Alternate Marking technique can be considered as Passive (e.g. RFC6374 Synonymous Flow Label or OAM Marking Bits in BIER Header).

The advantages of the method described in this document are:

- o easy implementation: it can be implemented or by using features already available on major routing platforms as described in Section 5.1 or by applying an optimized implementation of the method for both legacy and newest technologies;
- o low computational effort: the additional load on processing is negligible;
- o accurate packet loss measurement: single packet loss granularity is achieved with a passive measurement;
- o potential applicability to any kind of packet/frame -based traffic: Ethernet, IP, MPLS, etc., both unicast and multicast;
- o robustness: the method can tolerate out of order packets and it's not based on "special" packets whose loss could have a negative impact;
- o flexibility: all the timestamp formats are allowed, because they are managed out-of-band. The format (the Network Time Protocol (NTP) RFC 5905 [RFC5905] or the IEEE 1588 Precision Time Protocol (PTP) [IEEE-1588]) depends on the precision you want;
- o no interoperability issues: the features required to experiment and test the method (as described in Section 5.1) are available on all current routing platforms. Both a centralized or distributed solution can be used to harvest data from the routers.

The method doesn't raise any specific need for protocol extension, but it could be further improved by means of some extension to existing protocols. Specifically, the use of DiffServ bits for coloring the packets could not be a viable solution in some cases: a

standard method to color the packets for this specific application could be beneficial.

2. Overview of the method

In order to perform packet loss measurements on a production traffic flow, different approaches exist. The most intuitive one consists in numbering the packets, so that each router that receives the flow can immediately detect a packet missing. This approach, though very simple in theory, is not simple to achieve: it requires the insertion of a sequence number into each packet and the devices must be able to extract the number and check it in real time. Such a task can be difficult to implement on live traffic: if UDP is used as the transport protocol, the sequence number is not available; on the other hand, if a higher layer sequence number (e.g. in the RTP header) is used, extracting that information from each packet and process it in real time could overload the device.

An alternate approach is to count the number of packets sent on one end, the number of packets received on the other end, and to compare the two values. This operation is much simpler to implement, but requires that the devices performing the measurement are in sync: in order to compare two counters it is required that they refer exactly to the same set of packets. Since a flow is continuous and cannot be stopped when a counter has to be read, it can be difficult to determine exactly when to read the counter. A possible solution to overcome this problem is to virtually split the flow in consecutive blocks by inserting periodically a delimiter so that each counter refers exactly to the same block of packets. The delimiter could be for example a special packet inserted artificially into the flow. However, delimiting the flow using specific packets has some limitations. First, it requires generating additional packets within the flow and requires the equipment to be able to process those packets. In addition, the method is vulnerable to out of order reception of delimiting packets and, to a lesser extent, to their loss.

The method proposed in this document follows the second approach, but it doesn't use additional packets to virtually split the flow in blocks. Instead, it "marks" the packets so that the packets belonging to the same block will have the same color, whilst consecutive blocks will have different colors. Each change of color represents a sort of auto-synchronization signal that guarantees the consistency of measurements taken by different devices along the path (see also [I-D.cociglio-mboned-multicast-pm] and [I-D.tempia-opsawg-p3m], where this technique was introduced).

Figure 1 represents a very simple network and shows how the method can be used to measure packet loss on different network segments: by enabling the measurement on several interfaces along the path, it is possible to perform link monitoring, node monitoring or end-to-end monitoring. The method is flexible enough to measure packet loss on any segment of the network and can be used to isolate the faulty element.

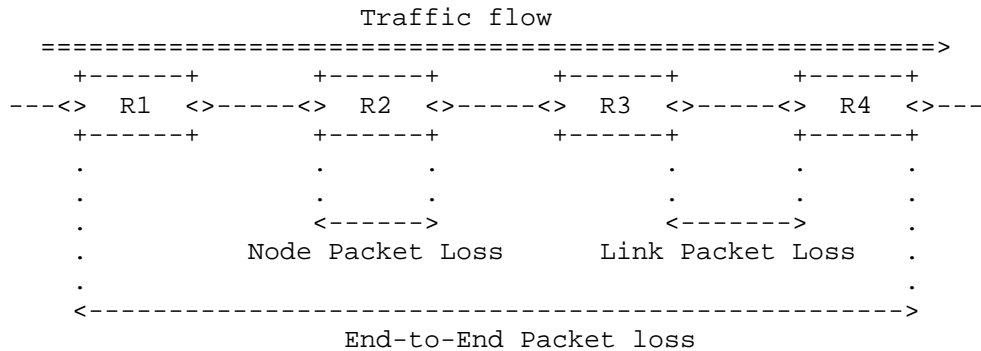


Figure 1: Available measurements

3. Detailed description of the method

This section describes in detail how the method operate. A special emphasis is given to the measurement of packet loss, that represents the core application of the method, but applicability to delay and jitter measurements is also considered.

3.1. Packet loss measurement

The basic idea is to virtually split traffic flows into consecutive blocks: each block represents a measurable entity unambiguously recognizable by all network devices along the path. By counting the number of packets in each block and comparing the values measured by different network devices along the path, it is possible to measure packet loss occurred in any single block between any two points.

As discussed in the previous section, a simple way to create the blocks is to "color" the traffic (two colors are sufficient) so that packets belonging to different consecutive blocks will have different colors. Whenever the color changes, the previous block terminates and the new one begins. Hence, all the packets belonging to the same block will have the same color and packets of different consecutive blocks will have different colors. The number of packets in each block depends on the criterion used to create the blocks:

- o if the color is switched after a fixed number of packets, then each block will contain the same number of packets (except for any losses);
- o if the color is switched according to a fixed timer, then the number of packets may be different in each block depending on the packet rate.

The following figure shows how a flow looks like when it is split in traffic blocks with colored packets.

A: packet with A coloring
 B: packet with B coloring

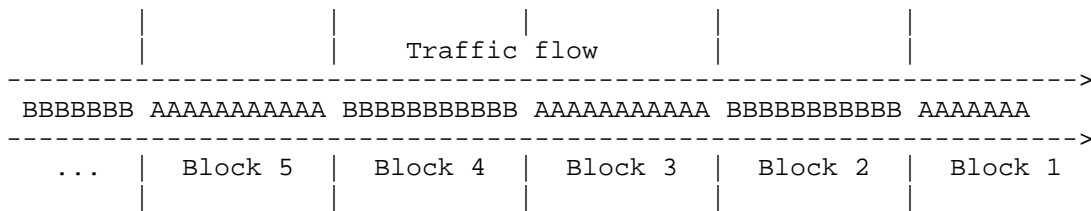


Figure 2: Traffic coloring

Figure 3 shows how the method can be used to measure link packet loss between two adjacent nodes.

Referring to the figure, let's assume we want to monitor the packet loss on the link between two routers: router R1 and router R2. According to the method, the traffic is colored alternatively with two different colors, A and B. Whenever the color changes, the transition generates a sort of square-wave signal, as depicted in the following figure.

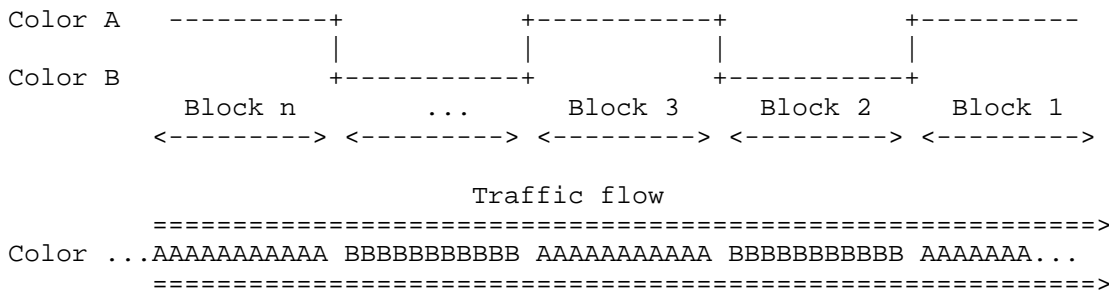


Figure 3: Computation of link packet loss

Traffic coloring could be done by R1 itself or it is already done before. R1 needs two counters, C(A)R1 and C(B)R1, on its egress interface: C(A)R1 counts the packets with color A and C(B)R1 counts those with color B. As long as traffic is colored A, only counter C(A)R1 will be incremented, while C(B)R1 is not incremented; vice versa, when the traffic is colored as B, only C(B)R1 is incremented. C(A)R1 and C(B)R1 can be used as reference values to determine the packet loss from R1 to any other measurement point down the path. Router R2, similarly, will need two counters on its ingress interface, C(A)R2 and C(B)R2, to count the packets received on that interface and colored with color A and B respectively. When an A block ends, it is possible to compare C(A)R1 and C(A)R2 and calculate the packet loss within the block; similarly, when the successive B block terminates, it is possible to compare C(B)R1 with C(B)R2, and so on for every successive block.

Likewise, by using two counters on R2 egress interface it is possible to count the packets sent out of R2 interface and use them as reference values to calculate the packet loss from R2 to any measurement point down R2.

Using a fixed timer for color switching offers a better control over the method: the (time) length of the blocks can be chosen large enough to simplify the collection and the comparison of measures taken by different network devices. It's preferable to read the value of the counters not immediately after the color switch: some packets could arrive out of order and increment the counter associated to the previous block (color), so it is worth waiting for some time. A safe choice is to wait $L/2$ time units (where L is the duration for each block) after the color switch, to read the still counter of the previous color, so the possibility to read a running counter instead of a still one is minimized. The drawback is that the longer the duration of the block, the less frequent the measurement can be taken.

The following table shows how the counters can be used to calculate the packet loss between R1 and R2. The first column lists the sequence of traffic blocks while the other columns contain the counters of A-colored packets and B-colored packets for R1 and R2. In this example, we assume that the values of the counters are reset to zero whenever a block ends and its associated counter has been read: with this assumption, the table shows only relative values, that is the exact number of packets of each color within each block. If the values of the counters were not reset, the table would contain cumulative values, but the relative values could be determined simply by difference from the value of the previous block of the same color.

The color is switched on the basis of a fixed timer (not shown in the table), so the number of packets in each block is different.

Block	C(A)R1	C(B)R1	C(A)R2	C(B)R2	Loss
1	375	0	375	0	0
2	0	388	0	388	0
3	382	0	381	0	1
4	0	377	0	374	3
...
2n	0	387	0	387	0
2n+1	379	0	377	0	2

Table 1: Evaluation of counters for packet loss measurements

During an A block (blocks 1, 3 and 2n+1), all the packets are A-colored, therefore the C(A) counters are incremented to the number seen on the interface, while C(B) counters are zero. Vice versa, during a B block (blocks 2, 4 and 2n), all the packets are B-colored: C(A) counters are zero, while C(B) counters are incremented.

When a block ends (because of color switching) the relative counters stop incrementing and it is possible to read them, compare the values measured on router R1 and R2 and calculate the packet loss within that block.

For example, looking at the table above, during the first block (A-colored), C(A)R1 and C(A)R2 have the same value (375), which corresponds to the exact number of packets of the first block (no loss). Also during the second block (B-colored) R1 and R2 counters have the same value (388), which corresponds to the number of packets of the second block (no loss). During blocks three and four, R1 and R2 counters are different, meaning that some packets have been lost: in the example, one single packet (382-381) was lost during block three and three packets (377-374) were lost during block four.

The method applied to R1 and R2 can be extended to any other router and applied to more complex networks, as far as the measurement is enabled on the path followed by the traffic flow(s) being observed.

It's worth mentioning two different strategies that can be used when implementing the method:

- o flow-based: the flow-based strategy is used when only a limited number of traffic flows need to be monitored. According to this strategy, only a subset of the flows is colored. Counters for packet loss measurements can be instantiated for each single flow, or for the set as a whole, depending on the desired granularity. A relevant problem with this approach is the necessity to know in advance the path followed by flows that are subject to measurement. Path rerouting and traffic load-balancing increase the issue complexity, especially for unicast traffic. The problem is easier to solve for multicast traffic where load balancing is seldom used and static joins are frequently used to force traffic forwarding and replication.
- o link-based: measurements are performed on all the traffic on a link by link basis. The link could be a physical link or a logical link. Counters could be instantiated for the traffic as a whole or for each traffic class (in case it is desired to monitor each class separately), but in the second case a couple of counters is needed for each class.

As mentioned, the flow-based measurement requires the identification of the flow to be monitored and the discovery of the path followed by the selected flow. It is possible to monitor a single flow or multiple flows grouped together, but in this case measurement is consistent only if all the flows in the group follow the same path. Moreover if a measurement is performed by grouping many flows, it is not possible to determine exactly which flow was affected by packets loss. In order to have measures per single flow it is necessary to configure counters for each specific flow. Once the flow(s) to be monitored have been identified, it is necessary to configure the monitoring on the proper nodes. Configuring the monitoring means configuring the rule to intercept the traffic and configuring the counters to count the packets. To have just an end-to-end monitoring, it is sufficient to enable the monitoring on the first and the last hop routers of the path: the mechanism is completely transparent to intermediate nodes and independent from the path followed by traffic flows. On the contrary, to monitor the flow on a hop-by-hop basis along its whole path it is necessary to enable the monitoring on every node from the source to the destination. In case the exact path followed by the flow is not known a priori (i.e. the flow has multiple paths to reach the destination) it is necessary to enable the monitoring system on every path: counters on interfaces traversed by the flow will report packet count, counters on other interfaces will be null.

3.1.1. Coloring the packets

The coloring operation is fundamental in order to create packet blocks. This implies choosing where to activate the coloring and how to color the packets.

In case of flow-based measurements, the flow to monitor can be defined by a set of selection rules (e.g. headers fields) used to match a subset of the packets; in this way it is possible to control the number of involved nodes, the path followed by the packets and the size of the flows. It is possible, in general, to have multiple coloring nodes or a single coloring node that is easier to manage and doesn't rise any risk of conflict. Coloring in multiple nodes can be done and the requirement is that the coloring must change periodically between the nodes according to the timing considerations in Section 3.2; so every node, that is designated as a measurement point along the path, should be able to identify unambiguously the colored packets. Furthermore [I-D.fioccola-ippm-multipoint-alt-mark] generalizes the coloring for multipoint to multipoint flow. In addition, it can be advantageous to color the flow as close as possible to the source because it allows an end-to-end measure if a measurement point is enabled on the last-hop router as well.

For link-based measurements, all traffic needs to be colored when transmitted on the link. If the traffic had already been colored, then it has to be re-colored because the color must be consistent on the link. This means that each hop along the path must (re-)color the traffic; the color is not required to be consistent along different links.

Traffic coloring can be implemented by setting a specific bit in the packet header and changing the value of that bit periodically. How to choose the marking field depends on the application and is out of scope here. However some applications are reported in Section 5.

3.1.2. Counting the packets

For flow-based measurements, assuming that the coloring of the packets is performed only by the source nodes, the nodes between source and destination (included) have to count the colored packets that they receive and forward: this operation can be enabled on every router along the path or only on a subset, depending on which network segment is being monitored (a single link, a particular metro area, the backbone, the whole path). Since the color switches periodically between two values, two counters (one for each value) are needed: one counter for packets with color A and one counter for packets with color B. For each flow (or group of flows) being monitored and for every interface where the monitoring is active, a couple of counters

is needed. For example, in order to monitor separately 3 flows on a router with 4 interfaces involved, 24 counters are needed (2 counters for each of the 3 flows on each of the 4 interfaces). Furthermore [I-D.fioccola-ippm-multipoint-alt-mark] generalizes the counting for multipoint to multipoint flow.

In case of link-based measurements the behaviour is similar except that coloring and counting operations are performed on a link by link basis at each endpoint of the link.

Another important aspect to take into consideration is when to read the counters: in order to count the exact number of packets of a block the routers must perform this operation when that block has ended: in other words, the counter for color A must be read when the current block has color B, in order to be sure that the value of the counter is stable. This task can be accomplished in two ways. The general approach suggests to read the counters periodically, many times during a block duration, and to compare these successive readings: when the counter stops incrementing means that the current block has ended and its value can be elaborated safely. Alternatively, if the coloring operation is performed on the basis of a fixed timer, it is possible to configure the reading of the counters according to that timer: for example, reading the counter for color A every period in the middle of the subsequent block with color B is a safe choice. A sufficient margin should be considered between the end of a block and the reading of the counter, in order to take into account any out-of-order packets.

3.1.3. Collecting data and calculating packet loss

The nodes enabled to perform performance monitoring collect the value of the counters, but they are not able to directly use this information to measure packet loss, because they only have their own samples. For this reason, an external Network Management System (NMS) can be used to collect and elaborate data and to perform packet loss calculation. The NMS compares the values of counters from different nodes and can calculate if some packets were lost (even a single packet) and also where packets were lost.

The value of the counters needs to be transmitted to the NMS as soon as it has been read. This can be accomplished by using SNMP or FTP and can be done in Push Mode or Polling Mode. In the first case, each router periodically sends the information to the NMS, in the latter case it is the NMS that periodically polls routers to collect information. In any case, the NMS has to collect all the relevant values from all the routers within one cycle of the timer.

it would be also possible to use a protocol to exchange values of counters between the two endpoints in order to let them perform the packet loss calculation for each traffic direction.

A possible approach for the performance measurement architecture is explained in [I-D.chen-ippm-coloring-based-ipfpm-framework], while [I-D.chen-ippm-ipfpm-report] introduces new information elements of IPFIX (RFC 7011 [RFC7011]).

3.2. Timing aspects

This document introduces two color switching method: one is based on fixed number of packet, the other is based on fixed timer. But the method based on fixed timer is preferable because is more deterministic, and will be considered in the rest of the document.

By considering the clock error between network devices R1 and R2, they must be synchronized to the same clock reference with an accuracy of +/- L/2 time units, where L is the time duration of the block. So each colored packet can be assigned to the right batch by each router. This is because the minimum time distance between two packets of the same color but belonging to different batches is L time units.

In practice, there are also out of order at batch boundaries, strictly related to the delay between measurement points. This means that, without considering clock error, we wait L/2 after color switching to be sure to take a still counter.

In summary we need to take into account two contributions: clock error between network devices and the interval we need to wait to avoid out of order because of network delay.

The following figure explains both issues.

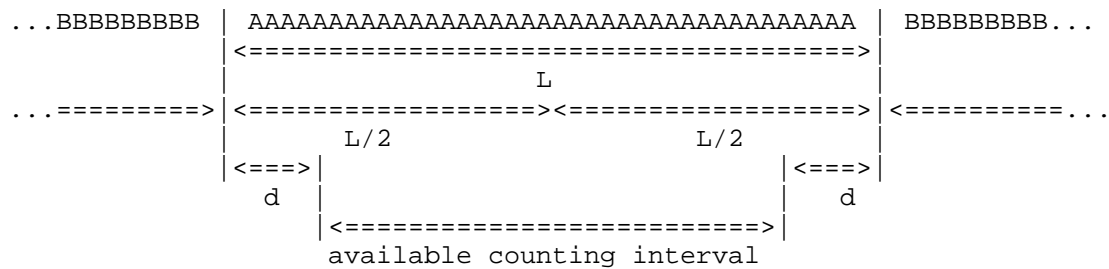


Figure 4: Timing aspects

It is assumed that all network devices are synchronized to a common reference time with an accuracy of $\pm A/2$. Thus, the difference between the clock values of any two network devices is bounded by A .

The guardband d is given by:

$$d = A + D_{\max} - D_{\min},$$

where A is the clock accuracy, D_{\max} is an upper bound on the network delay between the network devices, and D_{\min} is a lower bound on the delay.

The available counting interval is $L - 2d$ that must be > 0 .

The condition that must be satisfied and is a requirement on the synchronization accuracy is:

$$d < L/2.$$

3.3. One-way delay measurement

The same principle used to measure packet loss can be applied also to one-way delay measurement. There are three alternatives, as described hereinafter.

3.3.1. Single marking methodology

The alternation of colors can be used as a time reference to calculate the delay. Whenever the color changes (that means that a new block has started) a network device can store the timestamp of the first packet of the new block; that timestamp can be compared with the timestamp of the same packet on a second router to compute packet delay. Considering Figure 2, $R1$ stores a timestamp $TS(A)R1$ when it sends the first packet of block 1 (A-colored), a timestamp $TS(B)R1$ when it sends the first packet of block 2 (B-colored) and so on for every other block. $R2$ performs the same operation on the receiving side, recording $TS(A)R2$, $TS(B)R2$ and so on. Since the timestamps refer to specific packets (the first packet of each block) we are sure that timestamps compared to compute delay refer to the same packets. By comparing $TS(A)R1$ with $TS(A)R2$ (and similarly $TS(B)R1$ with $TS(B)R2$ and so on) it is possible to measure the delay between $R1$ and $R2$. In order to have more measurements, it is possible to take and store more timestamps, referring to other packets within each block.

In order to coherently compare timestamps collected on different routers, the clocks on the network nodes must be in sync. Furthermore, a measurement is valid only if no packet loss occurs and

if packet misordering can be avoided, otherwise the first packet of a block on R1 could be different from the first packet of the same block on R2 (f.i. if that packet is lost between R1 and R2 or it arrives after the next one).

The following table shows how timestamps can be used to calculate the delay between R1 and R2. The first column lists the sequence of blocks while other columns contain the timestamp referring to the first packet of each block on R1 and R2. The delay is computed as a difference between timestamps. For the sake of simplicity, all the values are expressed in milliseconds.

Block	TS(A)R1	TS(B)R1	TS(A)R2	TS(B)R2	Delay R1-R2
1	12.483	-	15.591	-	3.108
2	-	6.263	-	9.288	3.025
3	27.556	-	30.512	-	2.956
	-	18.113	-	21.269	3.156
...
2n	77.463	-	80.501	-	3.038
2n+1	-	24.333	-	27.433	3.100

Table 2: Evaluation of timestamps for delay measurements

The first row shows timestamps taken on R1 and R2 respectively and referring to the first packet of block 1 (which is A-colored). Delay can be computed as a difference between the timestamp on R2 and the timestamp on R1. Similarly, the second row shows timestamps (in milliseconds) taken on R1 and R2 and referring to the first packet of block 2 (which is B-colored). Comparing timestamps taken on different nodes in the network and referring to the same packets (identified using the alternation of colors) it is possible to measure delay on different network segments.

For the sake of simplicity, in the above example a single measurement is provided within a block, taking into account only the first packet of each block. The number of measurements can be easily increased by considering multiple packets in the block: for instance, a timestamp could be taken every N packets, thus generating multiple delay measurements. Taking this to the limit, in principle the delay could

be measured for each packet, by taking and comparing the corresponding timestamps (possible but impractical from an implementation point of view).

3.3.1.1. Mean delay

As mentioned before, the method previously exposed for measuring the delay is sensitive to out of order reception of packets. In order to overcome this problem, a different approach has been considered: it is based on the concept of mean delay. The mean delay is calculated by considering the average arrival time of the packets within a single block. The network device locally stores a timestamp for each packet received within a single block: summing all the timestamps and dividing by the total number of packets received, the average arrival time for that block of packets can be calculated. By subtracting the average arrival times of two adjacent devices it is possible to calculate the mean delay between those nodes. When computing the mean delay, measurement error could be augmented by accumulating measurement error of a lot of packets. This method is robust to out of order packets and also to packet loss (only a small error is introduced). Moreover, it greatly reduces the number of timestamps (only one per block for each network device) that have to be collected by the management system. On the other hand, it only gives one measure for the duration of the block (f.i. 5 minutes), and it doesn't give the minimum, maximum and median delay values (RFC 6703 [RFC6703]). This limitation could be overcome by reducing the duration of the block (f.i. from 5 minutes to a few seconds), that implicates an highly optimized implementation of the method.

By summing the mean delays of the two directions of a path, it is also possible to measure the two-way mean delay (round-trip delay).

3.3.2. Double marking methodology

The Single marking methodology for one-way delay measurement is sensitive to out of order reception of packets. The first approach to overcome this problem is described before and is based on the concept of mean delay. But the limitation of mean delay is that it doesn't give information about the delay values distribution for the duration of the block. Additionally it may be useful to have not only the mean delay but also the minimum, maximum and median delay values and, in wider terms, to know more about the statistic distribution of delay values. So in order to have more information about the delay and to overcome out of order issues, a different approach can be introduced: it is based on double marking methodology.

Basically, the idea is to use the first marking to create the alternate flow and, within this colored flow, a second marking to select the packets for measuring delay/jitter. The first marking is needed for packet loss and mean delay measurement. The second marking creates a new set of marked packets that are fully identified over the network, so that a network device can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on a second router to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. But the frequency of the second marking must be not too high in order to avoid out of order issues. Between packets with the second marking there should be a security time gap (e.g. this gap could be, at the minimum, the mean network delay calculated with the previous methodology) to avoid out of order issues and also to have a number of measurement packets that is rate independent. If a second marking packet is lost, the delay measurement for the considered block is corrupted and should be discarded.

Mean delay is calculated on all the packets of a sample and is a simple computation to be performed for single marking method. In some cases the mean delay measure is not sufficient to characterize the sample, and more statistics of delay extent data are needed, e.g. percentiles, variance and median delay values. The conventional range (maximum-minimum) should be avoided for several reasons, including stability of the maximum delay due to the influence by outliers. RFC 5481 [RFC5481] Section 6.5 highlights how the 99.9th percentile of delay and delay variation is more helpful to performance planners. To overcome this drawback the idea is to couple the mean delay measure for the entire batch with double marking method, where a subset of batch packets are selected for extensive delay calculation by using a second marking. In this way it is possible to perform a detailed analysis on these double marked packets. Please note that there are classic algorithms for median and variance calculation, but are out of the scope of this document. The comparison between the mean delay for the entire batch and the mean delay on these double marked packets gives an useful information since it is possible to understand if the double marking measurements are actually representative of the delay trends.

3.4. Delay variation measurement

Similarly to one-way delay measurement (both for single marking and double marking), the method can also be used to measure the inter-arrival jitter. We refer to the definition in RFC 3393 [RFC3393]. The alternation of colors, for single marking method, can be used as a time reference to measure delay variations. In case of double marking, the time reference is given by the second marked packets.

Considering the example depicted in Figure 2, R1 stores a timestamp $TS(A)R1$ whenever it sends the first packet of a block and R2 stores a timestamp $TS(B)R2$ whenever it receives the first packet of a block. The inter-arrival jitter can be easily derived from one-way delay measurement, by evaluating the delay variation of consecutive samples.

The concept of mean delay can also be applied to delay variation, by evaluating the average variation of the interval between consecutive packets of the flow from R1 to R2.

4. Considerations

This section highlights some considerations about the methodology.

4.1. Synchronization

The Alternate Marking technique does not require a strong synchronization, especially for packet loss and two-way delay measurement. Only one-way delay measurement requires network devices to have synchronized clocks.

The color switching is the reference for all the network devices, and the only requirement to be achieved is that all network devices have to recognize the right batch along the path.

If the length of the measurement period is L time units, then all network devices must be synchronized to the same clock reference with an accuracy of $\pm L/2$ time units (without considering network delay). This level of accuracy guarantees that all network devices consistently match the color bit to the correct block. For example, if the color is toggled every second ($L = 1$ second), then clocks must be synchronized with an accuracy of ± 0.5 second to a common time reference.

This synchronization requirement can be satisfied even with a relatively inaccurate synchronization method. This is true for packet loss and two-way delay measurement, instead, for one-way delay measurement clock synchronization must be accurate.

Therefore, a system that uses only packet loss and two-way delay measurement does not require synchronization. This is because the value of the clocks of network devices does not affect the computation of the two-way delay measurement.

4.2. Data Correlation

Data Correlation is the mechanism to compare counters and timestamps for packet loss, delay and delay variation calculation. It could be performed in several ways depending on the alternate marking application and use case.

- o A possibility is to use a centralized solution using Network Management System (NMS) to correlate data;
- o Another possibility is to define a protocol based distributed solution, by defining a new protocol or by extending the existing protocols (e.g. RFC6374, TWAMP, OWAMP) in order to communicate the counters and timestamps between nodes.

In the following paragraphs an example data correlation mechanism is explained and could be use independently of the adopted solutions.

When data is collected on the upstream and downstream node, e.g., packet counts for packet loss measurement or timestamps for packet delay measurement, and periodically reported to or pulled by other nodes or NMS, a certain data correlation mechanism SHOULD be in use to help the nodes or NMS to tell whether any two or more packet counts are related to the same block of markers, or any two timestamps are related to the same marked packet.

The alternate marking method described in this document literally split the packets of the measured flow into different measurement blocks, in addition a Block Number could be assigned to each of such measurement block. The BN is generated each time a node reads the data (packet counts or timestamps), and is associated with each packet count and timestamp reported to or pulled by other nodes or NMS. The value of BN could be calculated as the modulo of the local time (when the data are read) and the interval of the marking time period.

When the nodes or NMS see, for example, same BNs associated with two packet counts from an upstream and a downstream node respectively, it considers that these two packet counts corresponding to the same block, i.e. that these two packet counts belong to the same block of markers from the upstream and downstream node. The assumption of this BN mechanism is that the measurement nodes are time synchronized. This requires the measurement nodes to have a certain time synchronization capability (e.g., the Network Time Protocol (NTP) RFC 5905 [RFC5905], or the IEEE 1588 Precision Time Protocol (PTP) [IEEE-1588]). Synchronization aspects are further discussed in Section 4.

4.3. Packet Re-ordering

Due to ECMP, packet re-ordering is very common in IP network. The accuracy of marking based PM, especially packet loss measurement, may be affected by packet re-ordering. Take a look at the following example:

Block	1	2	3	4	5	...
Node R1	AAAAAAA	BBBBBBB	AAAAAAA	BBBBBBB	AAAAAAA	...
Node R2	AAAAABB	AABBBBA	AAABAAA	BBBBBBA	ABAAABA	...

Figure 5: Packet Reordering

In Figure 5 the packet stream for Node R1 isn't being reordered, and can be safely assigned to interval blocks, but the packet stream for Node R2 is being reordered, so, looking at the packet with the marker of "B" in block 3, there is no safe way to tell whether the packet belongs to block 2 or block 4.

In general there is the need to assign packets with the marker of "B" or "A" to the right interval blocks. Most of the packet re-ordering occur at the edge of adjacent blocks, and they are easy to handle if the interval of each block is sufficient large. Then, it can assume that the packets with different marker belong to the block that they are more close to. If the interval is small, it is difficult and sometime impossible to determine to which block a packet belongs.

To choose a proper interval is important and how to choose a proper interval is out of the scope of this document. But an implementation SHOULD provide a way to configure the interval and allow a certain degree of packet re-ordering.

5. Applications, implementation and deployment

The methodology described in the previous sections can be applied in various situations. Basically Alternate Marking technique could be used in many cases for performance measurement. The only requirement is to select and mark the flow to be monitored; in this way packets are batched by the sender and each batch is alternately marked such that can be easily recognized by the receiver.

Some recent alternate marking method applications are listed below:

- o IP flow performance measurement (IPFPM): this application of marking method is described in [I-D.chen-ippm-coloring-based-ipfpm-framework]. As an example, in this document, the last reserved bit of the Flag field of the IPv4

header is proposed to be used for marking, while a solution for IPv6 could be to leverage the IPv6 extension header for marking.

- o OAM Passive Performance Measurement: In [I-D.ietf-bier-mpls-encapsulation] two OAM bits from Bit Index Explicit Replication (BIER) Header are reserved for the passive performance measurement marking method. [I-D.ietf-bier-pmmm-oam] details the measurement for multicast service over BIER domain. In addition, the alternate marking method could also be used in a Service Function Chaining (SFC) domain. Lastly the application of the marking method to Network Virtualization Overlays (NVO3) protocols is considered by [I-D.ietf-nvo3-encap].
- o RFC6374 Use Case: RFC6374 [RFC6374] uses the LM packet as the packet accounting demarcation point. Unfortunately this gives rise to a number of problems that may lead to significant packet accounting errors in certain situations. [I-D.ietf-mpls-flow-ident] discusses the desired capabilities for MPLS flow identification in order to perform a better in-band performance monitoring of user data packets. A method of accomplishing identification is Synonymous Flow Labels (SFL) introduced in [I-D.bryant-mpls-sfl-framework], while [I-D.ietf-mpls-rfc6374-sfl] describes RFC6374 performance measurements with SFL.
- o active performance measurement: [I-D.fioccola-ippm-alt-mark-active] describes how to extend the existing Active Measurement Protocol, in order to implement alternate marking methodology. [I-D.fioccola-ippm-rfc6812-alt-mark-ext] describes an extension to the Cisco SLA Protocol Measurement-Type UDP-Measurement.

An example of implementation and deployment is explained in the next section, just to clarify how the method can work.

5.1. Report on the operational experiment

The method described in this document, also called PNP (Packet Network Performance Monitoring), has been invented and engineered in Telecom Italia.

It is important to highlight that the general description of the methodology in this document is a consequence of the operational experiment. The foundational elements of the technique have been tested and the lessons learnt from the operational experiment inspired the formalization of the Alternate Marking Method as detailed in the previous sections.

The methodology is experimented in Telecom Italia's network and is applied to multicast IPTV channels or other specific traffic flows with high QoS requirements (i.e. Mobile Backhauling traffic realized with a VPN MPLS).

This technology has been employed by leveraging functions and tools available on IP routers and it's currently being used to monitor packet loss in some portions of the Telecom Italia's network. The application of the method to delay measurement has also been evaluated in Telecom Italia's labs.

This Section describes how the experiment has been executed, in particular how the features currently available on existing routing platforms can be used to apply the method, in order to give an example of implementation and deployment.

The operational test, here described, uses the flow-based strategy, as defined in Section 3. Instead the link-based strategy could be applied to physical link or a logical link (e.g. Ethernet VLAN or a MPLS PW).

The implementation of the method leverages the available router functions, since the experiment has been done by a Service Provider (as Telecom Italia is) on its own network. So, with current router implementations, only QoS related fields and features offer the required flexibility to set bits in the packet header. In case a Service Provider only uses the three most significant bits of the DSCP field (corresponding to IP Precedence) for QoS classification and queuing, it is possible to use the two less significant bits of the DSCP field (bit 0 and bit 1) to implement the method without affecting QoS policies. That is the approach used for the experiment. One of the two bits (bit 0) could be used to identify flows subject to traffic monitoring (set to 1 if the flow is under monitoring, otherwise it is set to 0), while the second (bit 1) can be used for coloring the traffic (switching between values 0 and 1, corresponding to color A and B) and creating the blocks.

The experiment considers a flow as all the packets sharing the same source IP address or the same destination IP address, depending on the direction. In practice, once the flow has been defined, coloring the traffic using the DSCP field can be implemented by configuring on the router output interface an access list that intercepts the flow(s) to be monitored and applies to them a policy that sets the DSCP field accordingly. Since traffic coloring has to be switched between the two values over time, the policy needs to be modified periodically: an automatic script is used to perform this task on the basis of a fixed timer. The automatic script is loaded on board of

the router and automatizes the basic operations that are needed to realize the methodology.

After the traffic is colored using the DSCP field, all the routers on the path can perform the counting. For this purpose an access-list that matches specific DSCP values can be used to count the packets of the flow(s) being monitored. The same access-list can be installed on all the routers of the path. In addition, network flow monitoring, such as provided by IPFIX (RFC 7011 [RFC7011]), can be used to recognize timestamps of first/last packet of a batch in order to enable one of the alternatives to measure the delay as detailed in Section 3.3.

In the Telecom Italia's experiment the timer is set to 5 minutes, so the sequence of actions of the script is also executed every 5 minutes. This value has showed to be a good compromise between measurement frequency and stability of the measurement (i.e. possibility to collect all the measures referring to the same block).

For this experiment, both counters and any other data are collected by using the automatic script that sends out these to a Network Management System (NMS). The NMS is responsible for packet loss calculation, performed by comparing the values of counters from the routers along the flow(s) path. 5 minutes timer for color switching is a safe choice for reading the counters and is also coherent with the reporting window of the NMS.

Note that the use of the DSCP field for marking implies that the method in this case works reliably only within a single management and operation domain.

Lastly, the Telecom Italia experiment scales up to 1000 flows monitored together on a single router, while an implementation on dedicated hardware scales more, but it was tested only in labs for now.

5.1.1.1. Metric transparency

Since a Service Provider application is described here, the method can be applied to end-to-end services supplied to Customers. So it is important to highlight that the method MUST be transparent outside the Service Provider domain.

In Telecom Italia's implementation the source node colors the packets with a policy that is modified periodically via an automatic script in order to alternate the DSCP field of the packets. The nodes between source and destination (included) have to count with an access-list the colored packets that they receive and forward.

Moreover the destination node has an important role: the colored packets are intercepted and a policy restores and sets the DSCP field of all the packets to the initial value. In this way the metric is transparent because outside the section of the network under monitoring the traffic flow is unchanged.

In such a case, thanks to this restoring technique, network elements outside the Alternate Marking monitoring domain (e.g. the two Provider Edge nodes of the Mobile Backhauling VPN MPLS) are totally unaware that packets were marked. So this restoring technique makes Alternate Marking completely transparent outside its monitoring domain.

6. Hybrid measurement

The method has been explicitly designed for passive measurements but it can also be used with active measurements. In order to have both end to end measurements and intermediate measurements (hybrid measurements) two end points can exchanges artificial traffic flows and apply alternate marking over these flows. In the intermediate points artificial traffic is managed in the same way as real traffic and measured as specified before. So the application of marking method can simplify also the active measurement, as explained in [I-D.fioccola-ippm-alt-mark-active].

7. Compliance with RFC6390 guidelines

RFC6390 [RFC6390] defines a framework and a process for developing Performance Metrics for protocols above and below the IP layer (such as IP-based applications that operate over reliable or datagram transport protocols).

This document doesn't aim to propose a new Performance Metric but a new method of measurement for a few Performance Metrics that have already been standardized. Nevertheless, it's worth applying [RFC6390] guidelines to the present document, in order to provide a more complete and coherent description of the proposed method. We used a subset of the Performance Metric Definition template defined by [RFC6390].

- o Metric name and description: as already stated, this document doesn't propose any new Performance Metric. On the contrary, it describes a novel method for measuring packet loss [RFC7680]. The same concept, with small differences, can also be used to measure delay [RFC7679], and jitter [RFC3393]. The document mainly describes the applicability to packet loss measurement.

- o Method of Measurement or Calculation: according to the method described in the previous sections, the number of packets lost is calculated by subtracting the value of the counter on the source node from the value of the counter on the destination node. Both counters must refer to the same color. The calculation is performed when the value of the counters is in a steady state. The steady state is an intrinsic characteristic of the marking method counters because the alternation of color makes the counters associated to each color still one at a time for the duration of a marking period.
- o Units of Measurement: the method calculates and reports the exact number of packets sent by the source node and not received by the destination node.
- o Measurement Points: the measurement can be performed between adjacent nodes, on a per-link basis, or along a multi-hop path, provided that the traffic under measurement follows that path. In case of a multi-hop path, the measurements can be performed both end-to-end and hop-by-hop.
- o Measurement Timing: the method have a constraint on the frequency of measurements. This is detailed in Section 3.2, where it is specified that the marking period and the guardband interval are strictly related each other to avoid out of order issues. That is because, in order to perform a measure, the counter must be in a steady state and this happens when the traffic is being colored with the alternate color. As an example in the experiment of the method the time interval is set to 5 minutes, while other optimized implementations can also use a marking period of a few seconds.
- o Implementation: the experiment of the method uses two encodings of the DSCP field to color the packets; this enables the use of policy configurations on the router to color the packets and accordingly configure the counter for each color. The path followed by traffic being measured should be known in advance in order to configure the counters along the path and be able to compare the correct values.
- o Verification: both in the Lab and in the operational network the methodology has been tested and experimented for packet loss and delay measurements by using traffic generators together with precision test instruments and network emulators.
- o Use and Applications: the method can be used to measure packet loss with high precision on live traffic; moreover, by combining

end-to-end and per-link measurements, the method is useful to pinpoint the single link that is experiencing loss events.

- o Reporting Model: the value of the counters has to be sent to a centralized management system that perform the calculations; such samples must contain a reference to the time interval they refer to, so that the management system can perform the correct correlation; the samples have to be sent while the corresponding counter is in a steady state (within a time interval), otherwise the value of the sample should be stored locally.
- o Dependencies: the values of the counters have to be correlated to the time interval they refer to; moreover, as far the experiment of the method is based on DSCP values, there are significant dependencies on the usage of the DSCP field: it must be possible to rely on unused DSCP values without affecting QoS-related configuration and behavior; moreover, the intermediate nodes must not change the value of the DSCP field not to alter the measurement.
- o Organization of Results: the method of measurement produces singletons.
- o Parameters: currently, the main parameter of the method is the time interval used to alternate the colors and read the counters.

8. Security Considerations

This document specifies a method to perform measurements in the context of a Service Provider's network and has not been developed to conduct Internet measurements, so it does not directly affect Internet security nor applications which run on the Internet. However, implementation of this method 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.

- o Harm caused by the measurement: the measurements described in this document are passive, so there are no new packets injected into the network causing potential harm to the network itself and to data traffic. Nevertheless, the method implies modifications on the fly to a header or encapsulation of the data packets: this must be performed in a way that doesn't alter the quality of service experienced by packets subject to measurements and that preserve stability and performance of routers doing the measurements. One of the main security threats in OAM protocols is network reconnaissance; an attacker can gather information

about the network performance by passively eavesdropping to OAM messages. The advantage of the methods described in this document is that the marking bits are the only information that is exchanged between the network devices. Therefore, passive eavesdropping to data plane traffic does not allow attackers to gain information about the network performance.

- o Harm to the measurement: the measurements could be harmed by routers altering the marking of the packets, or by an attacker injecting artificial traffic. Authentication techniques, such as digital signatures, may be used where appropriate to guard against injected traffic attacks. Since the measurement itself may be affected by routers (or other network devices) along the path of IP packets intentionally altering the value of marking bits of packets, as mentioned above, the mechanism specified in this document can be applied just in the context of a controlled domain, and thus the routers (or other network devices) are locally administered and this type of attack can be avoided. In addition, an attacker can't gain information about network performance from a single monitoring point, and must use synchronized monitoring points at multiple points on the path, because they have to do the same kind of measurement and aggregation that Service Providers using Alternate Marking must do.

The privacy concerns of network measurement are limited because the method only relies on information contained in the header or encapsulation without any release of user data. Although information in the header or encapsulation is metadata that can be used to compromise the privacy of users, the limited marking technique in this document seems unlikely to substantially increase the existing privacy risks from header or encapsulation metadata. It might be theoretically possible to modulate the marking to serve as a covert channel, but it would have a very low data rate if it is to avoid adversely affecting the measurement systems that monitor the marking.

Delay attacks are another potential threat in the context of this document. Delay measurement is performed using a specific packet in each block, marked by a dedicated color bit. Therefore, a man-in-the-middle attacker can selectively induce synthetic delay only to delay-colored packets, causing systematic error in the delay measurements. As discussed in previous sections, the methods described in this document rely on an underlying time synchronization protocol. Thus, by attacking the time protocol an attacker can potentially compromise the integrity of the measurement. A detailed discussion about the threats against time protocols and how to mitigate them is presented in RFC 7384 [RFC7384].

9. IANA Considerations

There are no IANA actions required.

10. Acknowledgements

The previous IETF drafts about this technique were:
[I-D.cociglio-mboned-multicast-pm] and [I-D.tempia-opsawg-p3m].

The authors would like to thank Alberto Tempia Bonda, Domenico Laforgia, Daniele Accetta and Mario Bianchetti for their contribution to the definition and the implementation of the method.

The authors would also thank Spencer Dawkins, Carlos Pignataro, Brian Haberman and Eric Vyncke for their assistance and their detailed and precious reviews.

11. References

11.1. Normative References

- [IEEE-1588] IEEE 1588-2008, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", July 2008.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.

- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [I-D.bryant-mpls-sfl-framework]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S., and G. Mirsky, "Synonymous Flow Label Framework", draft-bryant-mpls-sfl-framework-05 (work in progress), June 2017.
- [I-D.chen-ippm-coloring-based-ipfpm-framework]
Chen, M., Zheng, L., Mirsky, G., Fioccola, G., and T. Mizrahi, "IP Flow Performance Measurement Framework", draft-chen-ippm-coloring-based-ipfpm-framework-06 (work in progress), March 2016.
- [I-D.chen-ippm-ipfpm-report]
Chen, M., Zheng, L., and G. Mirsky, "IP Flow Performance Measurement Report", draft-chen-ippm-ipfpm-report-01 (work in progress), April 2016.
- [I-D.cociglio-mboned-multicast-pm]
Cociglio, M., Capello, A., Bonda, A., and L. Castaldelli, "A method for IP multicast performance monitoring", draft-cociglio-mboned-multicast-pm-01 (work in progress), October 2010.
- [I-D.fioccola-ippm-alt-mark-active]
Fioccola, G., Clemm, A., Bryant, S., Cociglio, M., Chandramouli, M., and A. Capello, "Alternate Marking Extension to Active Measurement Protocol", draft-fioccola-ippm-alt-mark-active-01 (work in progress), March 2017.
- [I-D.fioccola-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate Marking method for passive and hybrid performance monitoring", draft-fioccola-ippm-multipoint-alt-mark-01 (work in progress), October 2017.

- [I-D.fioccola-ippm-rfc6812-alt-mark-ext]
Fioccola, G., Clemm, A., Cociglio, M., Chandramouli, M., and A. Capello, "Alternate Marking Extension to Cisco SLA Protocol RFC6812", draft-fioccola-ippm-rfc6812-alt-mark-ext-01 (work in progress), March 2016.
- [I-D.ietf-bier-mpls-encapsulation]
Wijnands, I., Rosen, E., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication in MPLS and non-MPLS Networks", draft-ietf-bier-mpls-encapsulation-12 (work in progress), October 2017.
- [I-D.ietf-bier-pmmm-oam]
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-pmmm-oam-03 (work in progress), October 2017.
- [I-D.ietf-mpls-flow-ident]
Bryant, S., Pignataro, C., Chen, M., Li, Z., and G. Mirsky, "MPLS Flow Identification Considerations", draft-ietf-mpls-flow-ident-05 (work in progress), July 2017.
- [I-D.ietf-mpls-rfc6374-sfl]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S., Mirsky, G., and G. Fioccola, "RFC6374 Synonymous Flow Labels", draft-ietf-mpls-rfc6374-sfl-01 (work in progress), December 2017.
- [I-D.ietf-nvo3-encap]
Boutros, S., Ganga, I., Garg, P., Manur, R., Mizrahi, T., Mozes, D., Nordmark, E., Smith, M., Aldrin, S., and I. Bagdonas, "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-01 (work in progress), October 2017.
- [I-D.tempia-opsawg-p3m]
Capello, A., Cociglio, M., Castaldelli, L., and A. Bonda, "A packet based method for passive performance monitoring", draft-tempia-opsawg-p3m-04 (work in progress), February 2014.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.

- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://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, <<https://www.rfc-editor.org/info/rfc6703>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.

Authors' Addresses

Giuseppe Fioccola (editor)
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Alessandro Capello
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: alessandro.capello@telecomitalia.it

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Luca Castaldelli
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: luca.castaldelli@telecomitalia.it

Mach(Guoyi) Chen
Huawei Technologies

Email: mach.chen@huawei.com

Lianshu Zheng
Huawei Technologies

Email: vero.zheng@huawei.com

Greg Mirsky
ZTE
USA

Email: gregimirsky@gmail.com

Tal Mizrahi
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 14, 2020

A. Morton
AT&T Labs
M. Bagnulo
UC3M
P. Eardley
BT
K. D'Souza
AT&T Labs
September 11, 2019

Initial Performance Metrics Registry Entries
draft-ietf-ippm-initial-registry-12

Abstract

This memo defines the set of Initial Entries for the IANA Performance Metrics Registry. The set includes, UDP Round-trip Latency and Loss, Packet Delay Variation, DNS Response Latency and Loss, UDP Poisson One-way Delay and Loss, UDP Periodic One-way Delay and Loss, ICMP Round-trip Latency and Loss, and TCP round-trip Latency and Loss.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14[RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on March 14, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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	6
2. Scope	7
3. Registry Categories and Columns	7
4. UDP Round-trip Latency and Loss Registry Entries	8
4.1. Summary	9
4.1.1. ID (Identifier)	9
4.1.2. Name	9
4.1.3. URIs	9
4.1.4. Description	9
4.1.5. Change Controller	9
4.1.6. Version (of Registry Format)	9
4.2. Metric Definition	9
4.2.1. Reference Definition	10
4.2.2. Fixed Parameters	10
4.3. Method of Measurement	11
4.3.1. Reference Method	11
4.3.2. Packet Stream Generation	12
4.3.3. Traffic Filtering (observation) Details	13
4.3.4. Sampling Distribution	13
4.3.5. Run-time Parameters and Data Format	13
4.3.6. Roles	14
4.4. Output	14
4.4.1. Type	14
4.4.2. Reference Definition	14
4.4.3. Metric Units	15
4.4.4. Calibration	15
4.5. Administrative items	16
4.5.1. Status	16
4.5.2. Requestor	16
4.5.3. Revision	16
4.5.4. Revision Date	16

4.6.	Comments and Remarks	16
5.	Packet Delay Variation Registry Entry	16
5.1.	Summary	16
5.1.1.	ID (Identifier)	16
5.1.2.	Name	16
5.1.3.	URIs	17
5.1.4.	Description	17
5.1.5.	Change Controller	17
5.1.6.	Version (of Registry Format)	17
5.2.	Metric Definition	17
5.2.1.	Reference Definition	17
5.2.2.	Fixed Parameters	18
5.3.	Method of Measurement	18
5.3.1.	Reference Method	19
5.3.2.	Packet Stream Generation	19
5.3.3.	Traffic Filtering (observation) Details	20
5.3.4.	Sampling Distribution	20
5.3.5.	Run-time Parameters and Data Format	20
5.3.6.	Roles	21
5.4.	Output	21
5.4.1.	Type	21
5.4.2.	Reference Definition	21
5.4.3.	Metric Units	21
5.4.4.	Calibration	22
5.5.	Administrative items	22
5.5.1.	Status	22
5.5.2.	Requestor	23
5.5.3.	Revision	23
5.5.4.	Revision Date	23
5.6.	Comments and Remarks	23
6.	DNS Response Latency and Loss Registry Entries	23
6.1.	Summary	23
6.1.1.	ID (Identifier)	23
6.1.2.	Name	24
6.1.3.	URI	24
6.1.4.	Description	24
6.1.5.	Change Controller	24
6.1.6.	Version (of Registry Format)	24
6.2.	Metric Definition	24
6.2.1.	Reference Definition	24
6.2.2.	Fixed Parameters	25
6.3.	Method of Measurement	27
6.3.1.	Reference Method	27
6.3.2.	Packet Stream Generation	28
6.3.3.	Traffic Filtering (observation) Details	29
6.3.4.	Sampling Distribution	29
6.3.5.	Run-time Parameters and Data Format	29
6.3.6.	Roles	30

6.4.	Output	30
6.4.1.	Type	30
6.4.2.	Reference Definition	30
6.4.3.	Metric Units	31
6.4.4.	Calibration	31
6.5.	Administrative items	32
6.5.1.	Status	32
6.5.2.	Requestor	32
6.5.3.	Revision	32
6.5.4.	Revision Date	32
6.6.	Comments and Remarks	32
7.	UDP Poisson One-way Delay and Loss Registry Entries	32
7.1.	Summary	32
7.1.1.	ID (Identifier)	32
7.1.2.	Name	33
7.1.3.	URI and URL	33
7.1.4.	Description	33
7.2.	Metric Definition	34
7.2.1.	Reference Definition	34
7.2.2.	Fixed Parameters	35
7.3.	Method of Measurement	35
7.3.1.	Reference Method	36
7.3.2.	Packet Stream Generation	36
7.3.3.	Traffic Filtering (observation) Details	37
7.3.4.	Sampling Distribution	37
7.3.5.	Run-time Parameters and Data Format	37
7.3.6.	Roles	38
7.4.	Output	38
7.4.1.	Type	38
7.4.2.	Reference Definition	38
7.4.3.	Metric Units	41
7.4.4.	Calibration	41
7.5.	Administrative items	42
7.5.1.	Status	42
7.5.2.	Requestor	42
7.5.3.	Revision	42
7.5.4.	Revision Date	42
7.6.	Comments and Remarks	42
8.	UDP Periodic One-way Delay and Loss Registry Entries	42
8.1.	Summary	43
8.1.1.	ID (Identifier)	43
8.1.2.	Name	43
8.1.3.	URIs	43
8.1.4.	Description	43
8.2.	Metric Definition	44
8.2.1.	Reference Definition	44
8.2.2.	Fixed Parameters	45
8.3.	Method of Measurement	46

8.3.1.	Reference Method	46
8.3.2.	Packet Stream Generation	47
8.3.3.	Traffic Filtering (observation) Details	47
8.3.4.	Sampling Distribution	47
8.3.5.	Run-time Parameters and Data Format	48
8.3.6.	Roles	48
8.4.	Output	48
8.4.1.	Type	48
8.4.2.	Reference Definition	49
8.4.3.	Metric Units	51
8.4.4.	Calibration	52
8.5.	Administrative items	53
8.5.1.	Status	53
8.5.2.	Requestor	53
8.5.3.	Revision	53
8.5.4.	Revision Date	53
8.6.	Comments and Remarks	53
9.	ICMP Round-trip Latency and Loss Registry Entries	53
9.1.	Summary	53
9.1.1.	ID (Identifier)	53
9.1.2.	Name	54
9.1.3.	URIs	54
9.1.4.	Description	54
9.1.5.	Change Controller	54
9.1.6.	Version (of Registry Format)	54
9.2.	Metric Definition	55
9.2.1.	Reference Definition	55
9.2.2.	Fixed Parameters	55
9.3.	Method of Measurement	56
9.3.1.	Reference Method	56
9.3.2.	Packet Stream Generation	57
9.3.3.	Traffic Filtering (observation) Details	58
9.3.4.	Sampling Distribution	58
9.3.5.	Run-time Parameters and Data Format	58
9.3.6.	Roles	59
9.4.	Output	59
9.4.1.	Type	59
9.4.2.	Reference Definition	59
9.4.3.	Metric Units	61
9.4.4.	Calibration	61
9.5.	Administrative items	62
9.5.1.	Status	62
9.5.2.	Requestor	62
9.5.3.	Revision	62
9.5.4.	Revision Date	62
9.6.	Comments and Remarks	62
10.	TCP Round-Trip Delay and Loss Registry Entries	62
10.1.	Summary	63

10.1.1.	ID (Identifier)	63
10.1.2.	Name	63
10.1.3.	URIs	63
10.1.4.	Description	63
10.1.5.	Change Controller	64
10.1.6.	Version (of Registry Format)	64
10.2.	Metric Definition	64
10.2.1.	Reference Definitions	64
10.2.2.	Fixed Parameters	66
10.3.	Method of Measurement	67
10.3.1.	Reference Methods	67
10.3.2.	Packet Stream Generation	69
10.3.3.	Traffic Filtering (observation) Details	69
10.3.4.	Sampling Distribution	69
10.3.5.	Run-time Parameters and Data Format	69
10.3.6.	Roles	70
10.4.	Output	70
10.4.1.	Type	70
10.4.2.	Reference Definition	70
10.4.3.	Metric Units	72
10.4.4.	Calibration	72
10.5.	Administrative items	73
10.5.1.	Status	73
10.5.2.	Requestor	73
10.5.3.	Revision	73
10.5.4.	Revision Date	73
10.6.	Comments and Remarks	73
11.	Security Considerations	73
12.	IANA Considerations	73
13.	Acknowledgements	73
14.	References	74
14.1.	Normative References	74
14.2.	Informative References	76
	Authors' Addresses	77

1. Introduction

This memo proposes an initial set of entries for the Performance Metrics Registry. It uses terms and definitions from the IPPM literature, primarily [RFC2330].

Although there are several standard templates for organizing specifications of performance metrics (see [RFC7679] 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. While examining aspects of metric

specifications which need to be registered, it became clear that none of the existing metric templates fully satisfies the particular needs of a registry.

Therefore, [I-D.ietf-ippm-metric-registry] defines the overall format for a Performance Metrics Registry. Section 5 of [I-D.ietf-ippm-metric-registry] also gives guidelines for those requesting registration of a Metric, that is the creation of entry(s) in the Performance Metrics Registry: "In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose." The process in [I-D.ietf-ippm-metric-registry] also requires that new entries are administered by IANA through Expert Review or IETF Standards action, which will ensure that the metrics are tightly defined.

2. Scope

This document defines the initial set of Performance Metrics Registry entries, for which IETF approval (following development in the IP Performance Metrics (IPPM) Working Group) will satisfy the requirement for Expert Review. Most are Active Performance Metrics, which are based on RFCs prepared in the IPPM working group of the IETF, according to their framework [RFC2330] and its updates.

3. Registry Categories and Columns

This memo uses the terminology defined in [I-D.ietf-ippm-metric-registry].

This section provides the categories and columns of the registry, for easy reference. An entry (row) therefore gives a complete description of a Registered Metric.

Registry Categories and Columns, shown as

Category	
Column	Column

Summary

Identifier	Name	URIs	Desc.	Reference	Change Controller	Ver
------------	------	------	-------	-----------	-------------------	-----

Metric Definition

Reference Definition	Fixed Parameters
----------------------	------------------

Method of Measurement

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

Output

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

Administrative Information

Status	Request	Rev	Rev.Date
--------	---------	-----	----------

Comments and Remarks

4. UDP Round-trip Latency and Loss Registry Entries

This section specifies an initial registry entry for the UDP Round-trip Latency, and another entry for UDP Round-trip Loss Ratio.

Note: Each Registry entry only produces a "raw" output or a statistical summary. To describe both "raw" and one or more statistics efficiently, the Identifier, Name, and Output Categories can be split and a single section can specify two or more closely-related metrics. This section specifies two Registry entries with many common columns. See Section 7 for an example specifying multiple Registry entries with many common columns.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign a corresponding URL to each Named Metric.

4.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

4.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

4.1.2. Name

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsec4_Seconds_95Percentile

RTLoss_Active_IP-UDP-Periodic_RFCXXXXsec4_Percent_LossRatio

4.1.3. URIs

URL: <http://<TBD by IANA>/<name>>

4.1.4. Description

RTDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the 95th percentile of their conditional delay distribution.

RTLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

4.1.5. Change Controller

IETF

4.1.6. Version (of Registry Format)

1.0

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

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

[RFC2681]

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-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" 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).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

4.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL: set to 255
 - * Protocol: Set to 17 (UDP)
- o IPv6 header values:

- * DSCP: set to 0
- * Hop Count: set to 255
- * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and included in the header
- o UDP Payload
 - * total of 100 bytes

Other measurement parameters:

- o Tmax: a loss threshold waiting time
 - * 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

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

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 Tmax defined under Fixed Parameters. However, the Periodic stream will be generated according to [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process

which calculates the RTT value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

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 MUST be included in the method of measurement for this metric.

4.3.2. Packet Stream Generation

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.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

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).

NA

4.3.4. Sampling Distribution

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.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

4.3.6. Roles

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

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, 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.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of Round-trip delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton Round-trip delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

4.4.2. Reference Definition

For all outputs ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalPkts the count of packets sent by the Src to Dst during the measurement interval.

For

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsec4_Seconds_95Percentile:

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as

For

RTLoss_Active_IP-UDP-Periodic_RFCXXXXsec4_Percent_LossRatio:

Percentile The numeric value of the result is expressed in units of lost packets to total packets times 100%, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.0000000001.

4.4.3. Metric Units

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

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

4.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

4.5. Administrative items

4.5.1. Status

Current

4.5.2. Requestor

This RFC numner

4.5.3. Revision

1.0

4.5.4. Revision Date

YYYY-MM-DD

4.6. Comments and Remarks

None.

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.

5.1.1. ID (Identifier)

<insert numeric identifier, an integer>

5.1.2. Name

OWPDV_Active_IP-UDP-Periodic_RFCXXXXsec5_Seconds_95Percentile

5.1.3. URIs

URL: `http://<TBD by IANA>/<name>`

5.1.4. Description

An assessment of packet delay variation with respect to the minimum delay observed on the periodic stream, and the Output is expressed as the 95th percentile of the packet delay variation distribution.

5.1.5. Change Controller

IETF

5.1.6. Version (of Registry Format)

1.0

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

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]

See sections 2.4 and 3.4 of [RFC3393]. Singleton delay differences measured are referred to by the variable name "ddT" (applicable to all forms of delay variation). However, this metric entry specifies the PDV form defined in section 4.2 of [RFC5481], where the singleton PDV for packet *i* is referred to by the variable name "PDV(*i*)".

5.2.2. Fixed Parameters

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL: set to 255
 - * Protocol: Set to 17 (UDP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and included in the header
- o UDP Payload
 - * total of 200 bytes

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

F a selection function unambiguously defining the packets from the stream selected for the metric. See section 4.2 of [RFC5481] for the PDV form.

See the Packet Stream generation category for two additional Fixed Parameters.

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

See section 2.6 and 3.6 of [RFC3393] for general singleton element calculations. This metric entry requires implementation of the PDV form defined in section 4.2 of [RFC5481]. Also see measurement considerations in section 8 of [RFC5481].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

5.3.2. Packet Stream Generation

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.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a

positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

5.3.3. Traffic Filtering (observation) Details

NA

5.3.4. Sampling Distribution

NA

5.3.5. Run-time Parameters and Data Format

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

5.3.6. Roles

5.4. Output

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

5.4.1. Type

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, 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.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way PDV for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton one-way PDV values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

5.4.2. Reference Definition

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

5.4.3. Metric Units

The 95th Percentile of one-way PDV is expressed in seconds.

5.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.00000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

5.5. Administrative items

5.5.1. Status

Current

5.5.2. Requestor

This RFC number

5.5.3. Revision

1.0

5.5.4. Revision Date

YYYY-MM-DD

5.6. Comments and Remarks

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

6. DNS Response Latency and Loss Registry Entries

This section gives initial registry entries for DNS Response Latency and Loss from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different names. RFC 2681 [RFC2681] defines a Round-trip delay metric. We build on that metric by specifying several of the input parameters to precisely define two metrics for measuring DNS latency and loss.

Note to IANA: Each Registry "Name" below specifies a single registry entry, whose output format varies in accordance with the name.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

6.1. Summary

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

6.1.1. ID (Identifier)

<insert numeric identifier, an integer>

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

6.1.2. Name

RTDNS_Active_IP-UDP-Poisson_RFCXXXXsec6_Seconds_Raw

RLDNS_Active_IP-UDP-Poisson_RFCXXXXsec6_Logical_Raw

6.1.3. URI

URL: <http://<TBD by IANA>/<name>>

6.1.4. Description

This is a metric for DNS Response performance from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different resource names.

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

RLDNS: This metric indicates that the response was deemed lost. In other words, the response time exceeded the maximum waiting time.

6.1.5. Change Controller

IETF

6.1.6. Version (of Registry Format)

1.0

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

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]

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-singleton 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 [RFC2681] 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).

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both response time and loss metrics employ a maximum waiting time for received responses, so the count of lost packets to total packets sent is the basis for the loss determination as per Section 4.3 of [RFC6673].

6.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL set to 255
 - * Protocol: Set to 17 (UDP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:

- * Source port: 53
- * Destination port: 53
- * Checksum: the checksum must be calculated and included in the header
- o Payload: The payload contains a DNS message as defined in RFC 1035 [RFC1035] with the following values:
 - * The DNS header section contains:
 - + Identification (see the Run-time column)
 - + 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 Fully Qualified Domain Name (FQDN) provided as input for the test, see the Run-time column
 - + QTYPE: the query type provided as input for the test, see the Run-time column
 - + QCLASS: set to 1 for IN
 - * The other sections do not contain any Resource Records.

Other measurement parameters:

- o Tmax: a loss threshold waiting time (and to help disambiguate queries)
 - * 5.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

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

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

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 reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a response packet lost. Lost packets SHALL be designated as having undefined delay and counted for the RLDNS metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTT value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving reply.

DNS Messages bearing Queries provide for random ID Numbers in the Identification header field, so more than one query may be launched while a previous request is outstanding when the ID Number is used. Therefore, the ID Number MUST be retained at the Src or included with each response packet to disambiguate packet reordering if it occurs.

IF a DNS response does not arrive within Tmax, the response time RTDNS is undefined, and RLDNS = 1. The Message ID SHALL be used to disambiguate the successive queries that are otherwise identical.

Since the ID Number field is only 16 bits in length, it places a limit on the number of simultaneous outstanding DNS queries during a stress test from a single Src address.

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]. However, the DNS Server is expected to perform all required functions to prepare and send a response, so the response time will include processing time and network delay. Section 8 of [RFC6673] presents additional requirements which SHALL be included in the method of measurement for this metric.

In addition to operations described in [RFC2681], the Src MUST parse the DNS headers of the reply and prepare the information for subsequent reporting as a measured result, along with the Round-Trip Delay.

6.3.2. Packet Stream Generation

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.

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 Reciprocal_lambda = 1/lambda, in seconds.

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.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

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).

NA

6.3.4. Sampling Distribution

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.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

Reciprocal_lambda average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless

conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

Trunc Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value). (if fixed, Trunc = 30.0000 seconds.)

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 (Src) to match-up replies to outstanding queries.

QNAME The domain name of the Query, formatted as specified in section 4 of [RFC6991].

QTYPE The Query Type, which will correspond to the IP address family of the query (decimal 1 for IPv4 or 28 for IPv6, formatted as a uint16, as per section 9.2 of [RFC6020]).

6.3.6. Roles

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

Raw -- for each DNS Query packet sent, sets of values as defined in the next column, including the status of the response, only assigning delay values to successful query-response pairs.

6.4.2. Reference Definition

For all outputs:

T the time the DNS Query was sent during the measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339],

see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

dT The time value of the round-trip delay to receive the DNS response, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]. This value is undefined when the response packet is not received at Src within waiting time Tmax seconds.

Rcode The value of the Rcode field in the DNS response header, expressed as a uint64 as specified in section 9.2 of [RFC6020]. Non-zero values convey errors in the response, and such replies must be analyzed separately from successful requests.

6.4.3. Metric Units

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

RTLDNS: the Logical value, where 1 = Lost and 0 = Received.

6.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address and payload manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

6.5. Administrative items

6.5.1. Status

Current

6.5.2. Requestor

This RFC number

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. UDP Poisson One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Poisson One-way Delay, and one for UDP Poisson One-way Loss.

IANA Note: Registry "Name" below specifies a single registry entry, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

7.1. Summary

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

7.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the six Metrics.

7.1.2. Name

OWDelay_Active_IP-UDP-Poisson-
Payload250B_RFCXXXXsec7_Seconds_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss_Active_IP-UDP-Poisson-
Payload250B_RFCXXXXsec7_Percent_LossRatio

7.1.3. URI and URL

URL: <https://www.iana.org> \ ... <name>

7.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

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

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] 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].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

Almes, G., Kalidini, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", RFC 7680, DOI 10.17487/RFC7680, January 2016, <<http://www.rfc-editor.org/info/rfc7680>>.

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

7.2.2. Fixed Parameters

Type-P:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL: set to 255
 - * Protocol: Set to 17 (UDP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and included in the header
- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]
 - * Security features in use influence the number of Padding octets.
 - * 250 octets total, including the TWAMP format

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

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

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

7.3.2. Packet Stream Generation

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.

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. The reciprocal of lambda is the average packet spacing, thus the Run-time Parameter is $\text{Reciprocal_lambda} = 1/\text{lambda}$, in seconds.

Method 3 SHALL be 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 Parameter Trunc), and the Src sends each packet at the computed times.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

Reciprocal_lambda average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905]. Reciprocal_lambda = 1 packet per second.

Trunc Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value). Trunc = 30.0000 seconds.

7.3.3. Traffic Filtering (observation) Details

NA

7.3.4. Sampling Distribution

NA

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.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

7.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src. This is the TWAMP Session-Reflector.

7.4. Output

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

7.4.1. Type

See subsection titles below for Types.

7.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

7.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite 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 Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite 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 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay} [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.5. Std_Dev

The Std_Dev SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is the classic calculation for standard deviation of a population.

Std_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds.

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

7.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

time_offset The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

7.5. Administrative items

7.5.1. Status

Current

7.5.2. Requestor

This REFC number

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. UDP Periodic One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Periodic One-way Delay, and one for UDP Periodic One-way Loss.

IANA Note: Registry "Name" below specifies a single registry entry, whose output format varies according to the <statistic> element of

the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

8.1. Summary

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

8.1.1. ID (Identifier)

IANA is asked to assign a different numeric identifiers to each of the six Metrics.

8.1.2. Name

OWDelay_Active_IP-UDP-Periodic20m-
Payload142B_RFCXXXXsec8_Seconds_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss_Active_IP-UDP-Periodic-
Payload142B_RFCXXXXsec8_Percent_LossRatio

8.1.3. URIs

URL: <https://www.iana.org> \ ... <name>

8.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

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

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] 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].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

Almes, G., Kalidini, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", RFC 7680, DOI 10.17487/RFC7680, January 2016, <<http://www.rfc-editor.org/info/rfc7680>>.

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

8.2.2. Fixed Parameters

Type-P:

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL: set to 255
 - * Protocol: Set to 17 (UDP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Protocol: Set to 17 (UDP)
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and included in the header
- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]
 - * Security features in use influence the number of Padding octets.
 - * 142 octets total, including the TWAMP format (if used)

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

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

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters. However, a Periodic stream is used, as defined in [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format

of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

8.3.2. Packet Stream Generation

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.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200 expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

dT the duration of the interval for allowed sample start times, with value 1.0000, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

T0 the actual start time of the periodic stream, determined from T0 and dT.

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

These stream parameters will be specified as Run-time parameters.

8.3.3. Traffic Filtering (observation) Details

NA

8.3.4. Sampling Distribution

NA

8.3.5. 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.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

8.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src. This is the TWAMP Session-Reflector.

8.4. Output

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

8.4.1. Type

See subsection titles in Reference Definition for Latency Types.

8.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

8.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite 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 Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite 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 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.5. Std_Dev

The Std_Dev SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is the classic calculation for standard deviation of a population.

Std_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds, where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max

- o StdDev

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

8.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

8.5. Administrative items

8.5.1. Status

Current

8.5.2. Requestor

This RFC number

8.5.3. Revision

1.0

8.5.4. Revision Date

YYYY-MM-DD

8.6. Comments and Remarks

None.

9. ICMP Round-trip Latency and Loss Registry Entries

This section specifies three initial registry entries for the ICMP Round-trip Latency, and another entry for ICMP Round-trip Loss Ratio.

This section specifies four Registry entries with many common columns.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign four corresponding URLs to each Named Metric.

9.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

9.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

9.1.2. Name

RTDelay_Active_IP-ICMP-SendOnRcv_RFCXXXXsec9_Seconds_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss_Active_IP-ICMP-SendOnRcv_RFCXXXXsec9_Percent_LossRatio

9.1.3. URIs

URL: <http://<TBD by IANA>/<name>>

9.1.4. Description

RTDelay: This metric assesses the delay of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss: This metric assesses the loss ratio of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

9.1.5. Change Controller

IETF

9.1.6. Version (of Registry Format)

1.0

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.

9.2.1. Reference Definition

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

[RFC2681]

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-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" 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).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

9.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- * DSCP: set to 0

- * TTL: set to 255
- * Protocol: Set to 01 (ICMP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Limit: set to 255
 - * Protocol: Set to 01 (ICMP)
- o ICMP header values:
 - * Type: 8 (Echo Request)
 - * Code: 0
 - * Checksum: the checksum MUST be calculated and included in the header
 - * (Identifier and Sequence Number set at Run-Time)
- o ICMP Payload
 - * total of 32 bytes of random info

Other measurement parameters:

- o Tmax: a loss threshold waiting time
 - * 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

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.

9.3.1. Reference Method

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 Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTD) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTD value MAY enforce the Tmax threshold on stored values before calculations. 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.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

The measurement process will determine the sequence numbers applied to test packets after the Fixed and Runtime parameters are passed to that process. The ICMP measurement process and protocol will dictate the format of sequence numbers and other identifiers.

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 MUST be included in the method of measurement for this metric.

9.3.2. Packet Stream Generation

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.

The ICMP metrics use a sending discipline called "SendOnRcv" or Send On Receive. This is a modification of Section 3 of [RFC3432], which prescribes the method for generating Periodic streams using associated parameters as defined below for this description:

incT the nominal duration of inter-packet interval, first bit to first bit

dT the duration of the interval for allowed sample start times

The incT stream parameter will be specified as a Run-time parameter, and dT is not used in SendOnRcv.

A SendOnRcv sender behaves exactly like a Periodic stream generator while all reply packets arrive with $RTD < incT$, and the inter-packet interval will be constant.

If a reply packet arrives with $RTD \geq incT$, then the inter-packet interval for the next sending time is nominally RTD.

If a reply packet fails to arrive within Tmax, then the inter-packet interval for the next sending time is nominally Tmax.

If an immediate send on reply arrival is desired, then set $incT=0$.

9.3.3. 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).

NA

9.3.4. Sampling Distribution

NA

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.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

incT the nominal duration of inter-packet interval, first bit to first bit, expressed in units of seconds, as a positive value of

type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Count The total count of ICMP Echo Requests to send, formatted as a uint16, as per section 9.2 of [RFC6020].

(see the Packet Stream Generation section for additional Run-time parameters)

9.3.6. Roles

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.

9.4. Output

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

9.4.1. Type

See subsection titles in Reference Definition for Latency Types.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

9.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalCount the count of packets actually sent by the Src to Dst during the measurement interval.

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

9.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite 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 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay} [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

9.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface

contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

9.5. Administrative items

9.5.1. Status

Current

9.5.2. Requestor

This RFC number

9.5.3. Revision

1.0

9.5.4. Revision Date

YYYY-MM-DD

9.6. Comments and Remarks

None

10. TCP Round-Trip Delay and Loss Registry Entries

This section specifies three initial registry entries for the Passive assessment of TCP Round-Trip Delay (RTD) and another entry for TCP Round-trip Loss Count.

This section specifies four Registry entries with many common columns.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes

four closely-related registry entries. As a result, IANA is also asked to assign four corresponding URLs to each Named Metric.

10.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

10.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

10.1.2. Name

RTDelay_Passive_IP-TCP_RFCXXXXsec10_Seconds_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTDelay_Passive_IP-TCP-HS_RFCXXXXsec10_Seconds_Singleton

Note that a mid-point observer only has the opportunity to compose a single RTDelay on the TCP Hand Shake.

RTLoss_Passive_IP-TCP_RFCXXXXsec10_Packet_Count

10.1.3. URIs

URL: <https://<TBD by IANA>/<name>>

10.1.4. Description

RTDelay: This metric assesses the round-trip delay of TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean

- o Min
- o Max

RTLoss: This metric assesses the estimated loss count for TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the estimated Loss Count for the measurement interval.

10.1.5. Change Controller

IETF

10.1.6. Version (of Registry Format)

1.0

10.2. 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.

10.2.1. Reference Definitions

Although there is no RFC that describes passive measurement of Round-Trip Delay, the parallel definition for Active measurement is:

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

[RFC2681]

This metric definition uses the terms singleton and sample as defined in Section 11 of [RFC2330]. (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-singleton sample.)

With the Observation Point [RFC7011] (OP) typically located between the hosts participating in the TCP connection, the Round-trip Delay metric requires two individual measurements between the OP and each host, such that the Spatial Composition [RFC6049] of the measurements yields a Round-trip Delay singleton (we are extending the composition of one-way subpath delays to subpath round-trip delay).

Using the direction of TCP SYN transmission to anchor the nomenclature, host A sends the SYN and host B replies with SYN-ACK during connection establishment. The direction of SYN transfer is considered the Forward direction of transmission, from A through OP to B (Reverse is B through OP to A).

Traffic filters reduce the packet stream at the OP to a Qualified bidirectional flow packets.

In the definitions below, Corresponding Packets are transferred in different directions and convey a common value in a TCP header field that establishes correspondence (to the extent possible). Examples may be found in the TCP timestamp fields.

For a real number, RTD_{fwd} , \gg the Round-trip Delay in the Forward direction from OP to host B at time T' is $RTD_{fwd} \ll$ REQUIRES that OP observed a Qualified Packet to host B at wire-time T' , that host B received that packet and sent a Corresponding Packet back to host A, and OP observed the Corresponding Packet at wire-time $T' + RTD_{fwd}$.

For a real number, RTD_{rev} , \gg the Round-trip Delay in the Reverse direction from OP to host A at time T'' is $RTD_{rev} \ll$ REQUIRES that OP observed a Qualified Packet to host A at wire-time T'' , that host A received that packet and sent a Corresponding Packet back to host B, and that OP observed the Corresponding Packet at wire-time $T'' + RTD_{rev}$.

Ideally, the packet sent from host B to host A in both definitions above SHOULD be the same packet (or, when measuring RTD_{rev} first, the packet from host A to host B in both definitions should be the same).

The REQUIRED Composition Function for a singleton of Round-trip Delay at time T (where T is the earliest of T' and T'' above) is:

$$RTDelay = RTD_{fwd} + RTD_{rev}$$

Note that when OP is located at host A or host B, one of the terms composing $RTDelay$ will be zero or negligible.

When the Qualified and Corresponding Packets are a TCP-SYN and a TCP-SYN-ACK, then $RTD_{fwd} == RTD_{HS_{fwd}}$.

When the Qualified and Corresponding Packets are a TCP-SYN-ACK and a TCP-ACK, then $RTD_{rev} == RTD_{HS_{rev}}$.

The REQUIRED Composition Function for a singleton of Round-trip Delay for the connection Hand Shake:

$$\text{RTDelay_HS} = \text{RTD_HS_fwd} + \text{RTD_HS_rev}$$

The definition of Round-trip Loss Count uses the nomenclature developed above, based on observation of the TCP header sequence numbers and storing the sequence number gaps observed. Packet Losses can be inferred from:

- o Out-of-order segments: TCP segments are transmitted with monotonically increasing sequence numbers, but these segments may be received out of order. Section 3 of [RFC4737] describes the notion of "next expected" sequence numbers which can be adapted to TCP segments (for the purpose of detecting reordered packets). Observation of out-of-order segments indicates loss on the path prior to the OP, and creates a gap.
- o Duplicate segments: Section 2 of [RFC5560] defines identical packets and is suitable for evaluation of TCP packets to detect duplication. Observation of duplicate segments *without a corresponding gap* indicates loss on the path following the OP (because they overlap part of the delivered sequence numbers already observed at OP).

Each observation of an out-of-order or duplicate infers a singleton of loss, but composition of Round-trip Loss Counts will be conducted over a measurement interval which is synonymous with a single TCP connection.

With the above observations in the Forward direction over a measurement interval, the count of out-of-order and duplicate segments is defined as `RTL_fwd`. Comparable observations in the Reverse direction are defined as `RTL_rev`.

For a measurement interval (corresponding to a single TCP connection), T_0 to T_f , the REQUIRED Composition Function for a the two single-direction counts of inferred loss is:

$$\text{RTLoss} = \text{RTL_fwd} + \text{RTL_rev}$$

10.2.2. Fixed Parameters

Traffic Filters:

- o IPv4 header values:
 - * DSCP: set to 0
 - * Protocol: Set to 06 (TCP)

- o IPv6 header values:
 - * DSCP: set to 0
 - * Protocol: Set to 06 (TCP)
- o TCP header values:
 - * Flags: ACK, SYN, FIN, set as required
 - * Timestamp Option (TSopt): Set
 - + Section 3.2 of [RFC7323]

10.3. 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.

10.3.1. Reference Methods

The foundation methodology for this metric is defined in Section 4 of [RFC7323] using the Timestamp Option with modifications that allow application at a mid-path Observation Point (OP) [RFC7011]. Further details and applicable heuristics were derived from [Strowes] and [Trammell-14].

The Traffic Filter at the OP is configured to observe a single TCP connection. When the SYN, SYN-ACK, ACK handshake occurs, it offers the first opportunity to measure both RTD_fwd (on the SYN to SYN-ACK pair) and RTD_rev (on the SYN-ACK to ACK pair). Label this singleton of RTDelay as RTDelay_HS (composed using the forward and reverse measurement pair). RTDelay_HS SHALL be treated separately from other RTDelays on data-bearing packets and their ACKs. The RTDelay_HS value MAY be used as a sanity check on other Composed values of RTDelay.

For payload bearing packets, the OP measures the time interval between observation of a packet with Sequence Number *s*, and the corresponding ACK with same Sequence number. When the payload is transferred from host A to host B, the observed interval is RTD_fwd.

Because many data transfers are unidirectional (say, in the Forward direction from host A to host B), it is necessary to use pure ACK packets with Timestamp (TSval) and their Timestamp value echo to perform a RTD_rev measurement. The time interval between observation

of the ACK from B to A, and the corresponding packet with Timestamp echo (TSecr) is the RTD_rev.

Delay Measurement Filtering Heuristics:

If Data payloads were transferred in both Forward and Reverse directions, then the Round-Trip Time Measurement Rule in Section 4.1 of [RFC7323] could be applied. This rule essentially excludes any measurement using a packet unless it makes progress in the transfer (advances the left edge of the send window, consistent with[Strowes]).

A different heuristic from [Trammell-14] is to exclude any RTD_rev that is larger than previously observed values. This would tend to exclude Reverse measurements taken when the Application has no data ready to send, because considerable time could be added to RTD_rev from this source of error.

Note that the above Heuristic assumes that host A is sending data. Host A expecting a download would mean that this heuristic should be applied to RTD_fwd.

The statistic calculations to summarize the delay (RTDelay) SHALL be performed on the conditional distribution, conditioned on successful Forward and Reverse measurements which follow the Heuristics.

Method for Inferring Loss:

The OP tracks sequence numbers and stores gaps for each direction of transmission, as well as the next-expected sequence number as in [Trammell-14] and [RFC4737]. Loss is inferred from Out-of-order segments and Duplicate segments.

Loss Measurement Filtering Heuristics:

[Trammell-14] adds a window of evaluation based on the RTDelay.

Distinguish Re-ordered from OOO due to loss, because sequence number gap is filled during the same RTDelay window. Segments detected as re-ordered according to [RFC4737] MUST reduce the Loss Count inferred from Out-of-order segments.

Spurious (unneeded) retransmissions (observed as duplicates) can also be reduced this way, as described in [Trammell-14].

Sources of Error:

The principal source of RTDelay error is the host processing time to return a packet that defines the termination of a time interval. The heuristics above intend to mitigate these errors by excluding measurements where host processing time is a significant part of RTD_fwd or RTD_rev.

A key source of RTLoss error is observation loss, described in section 3 of [Trammell-14].

10.3.2. Packet Stream Generation

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.

NA

10.3.3. 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).

The Fixed Parameters above give a portion of the Traffic Filter. Other aspects will be supplied as Run-time Parameters (below).

10.3.4. Sampling Distribution

This metric requires a complete sample of all packets that qualify according to the Traffic Filter criteria.

10.3.5. 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.

Src the IP address of the host in the host A Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the host B (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3

of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Td is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Td Optionally, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]), or the duration (see T0). The UTC Time Zone is required by Section 6.1 of [RFC2330]. Alternatively, the end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK packets exchanged between host A and B effectively ends the interval.

TTL or Hop Limit Set at desired value.

10.3.6. Roles

host A launches the SYN packet to open the connection, and synonymous with an IP address.

host B replies with the SYN-ACK packet to open the connection, and synonymous with an IP address.

10.4. Output

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

10.4.1. Type

See subsection titles in Reference Definition for RTDelay Types.

For RTLoss -- the count of lost packets.

10.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. The end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK

packets exchanged between host A and B effectively ends the interval.

... ..

For RTDelay_HS -- the Round trip delay of the Handshake.

For RTLoss -- the count of lost packets.

For each <statistic>, one of the following sub-sections apply:

10.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite 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 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite 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 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Delay of the Hand Shake is expressed in seconds.

The Round-trip Loss Count is expressed as a number of packets.

10.4.4. Calibration

Passive measurements at an OP could be calibrated against an active measurement (with loss emulation) at host A or B, where the active measurement represents the ground-truth.

10.5. Administrative items

10.5.1. Status

Current

10.5.2. Requestor

This RFC

10.5.3. Revision

1.0

10.5.4. Revision Date

YYYY-MM-DD

10.6. Comments and Remarks

None.

11. Security Considerations

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

12. IANA Considerations

IANA is requested to populate The Performance Metrics Registry defined in [I-D.ietf-ippm-metric-registry] with the values defined in sections 4 through 10.

See the IANA Considerations section of [I-D.ietf-ippm-metric-registry] for additional requests and considerations.

13. 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 identifying the IPFIX metric with Flow Key as an example, for suggesting the Passive TCP RTD metric and supporting references, 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.

The authors also acknowledge the constructive reviews and helpful suggestions from Barbara Stark, Juergen Schoenwaelder, Tim Carey, Yaakov Stein, and participants in the LMAP working group. Thanks to Michelle Cotton for her early IANA review, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL.

14. References

14.1. 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, 2019.
- [RFC1035] Mockapetris, P., "Domain names - implementation and
specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://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,
<<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip
Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681,
September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
<<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation
Metric for IP Performance Metrics (IPPM)", RFC 3393,
DOI 10.17487/RFC3393, November 2002,
<<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network
performance measurement with periodic streams", RFC 3432,
DOI 10.17487/RFC3432, November 2002,
<<https://www.rfc-editor.org/info/rfc3432>>.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<https://www.rfc-editor.org/info/rfc4737>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5560] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, DOI 10.17487/RFC5560, May 2009, <<https://www.rfc-editor.org/info/rfc5560>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, DOI 10.17487/RFC6049, January 2011, <<https://www.rfc-editor.org/info/rfc6049>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <<https://www.rfc-editor.org/info/rfc7323>>.

- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [Strowes] Strowes, S., "Passively Measuring TCP Round Trip Times, Communications of the ACM, Vol. 56 No. 10, Pages 57-64", September 2013.
- [Trammell-14]
Trammell, B., "Inline Data Integrity Signals for Passive Measurement, In: Dainotti A., Mahanti A., Uhlig S. (eds) Traffic Monitoring and Analysis. TMA 2014. Lecture Notes in Computer Science, vol 8406. Springer, Berlin, Heidelberg https://link.springer.com/chapter/10.1007/978-3-642-54999-1_2", March 2014.

14.2. Informative References

- [RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <<https://www.rfc-editor.org/info/rfc1242>>.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://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, <<https://www.rfc-editor.org/info/rfc6703>>.

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

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

Kevin D'Souza
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Phone: +1 732 420 xxxx
Email: kld@att.com

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: March 14, 2020

M. Bagnulo
UC3M
B. Claise
Cisco Systems, Inc.
P. Eardley
BT
A. Morton
AT&T Labs
A. Akhter
Consultant
September 11, 2019

Registry for Performance Metrics
draft-ietf-ippm-metric-registry-20

Abstract

This document defines the format for the IANA Performance Metrics Registry. This document also gives a set of guidelines for Registered Performance Metric requesters and reviewers.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 14, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. Scope	6
4. Motivation for a Performance Metrics Registry	7
4.1. Interoperability	7
4.2. Single point of reference for Performance Metrics	8
4.3. Side benefits	8
5. Criteria for Performance Metrics Registration	9
6. Performance Metric Registry: Prior attempt	9
6.1. Why this Attempt Will Succeed	10
7. Definition of the Performance Metric Registry	11
7.1. Summary Category	12
7.1.1. Identifier	12
7.1.2. Name	13
7.1.3. URIs	16
7.1.4. Description	17
7.1.5. Reference	17
7.1.6. Change Controller	17
7.1.7. Version (of Registry Format)	17
7.2. Metric Definition Category	17
7.2.1. Reference Definition	17
7.2.2. Fixed Parameters	18
7.3. Method of Measurement Category	18
7.3.1. Reference Method	18
7.3.2. Packet Stream Generation	19
7.3.3. Traffic Filter	19
7.3.4. Sampling Distribution	20
7.3.5. Run-time Parameters	20
7.3.6. Role	21
7.4. Output Category	21
7.4.1. Type	22
7.4.2. Reference Definition	22
7.4.3. Metric Units	22
7.4.4. Calibration	22
7.5. Administrative information	23
7.5.1. Status	23
7.5.2. Requester	23
7.5.3. Revision	23
7.5.4. Revision Date	23
7.6. Comments and Remarks	23

8.	The Life-Cycle of Registered Performance Metrics	24
8.1.	Adding new Performance Metrics to the Performance Metrics Registry	24
8.2.	Revising Registered Performance Metrics	25
8.3.	Deprecating Registered Performance Metrics	27
9.	Security considerations	27
10.	IANA Considerations	28
10.1.	Registry Group	28
10.2.	Performance Metric Name Elements	28
10.3.	New Performance Metrics Registry	29
11.	Acknowledgments	30
12.	References	30
12.1.	Normative References	30
12.2.	Informative References	31
	Authors' Addresses	33

1. Introduction

The IETF specifies and uses Performance Metrics of protocols and applications transported over its protocols. Performance metrics are such an important part of the operations of IETF protocols that [RFC6390] specifies guidelines for their development.

The definition and use of Performance Metrics in the IETF happens in various working groups (WG), most notably:

The "IP Performance Metrics" (IPPM) WG is the WG primarily focusing on Performance Metrics definition at the IETF.

The "Metric Blocks for use with RTCP's Extended Report Framework" (XRBLOCK) WG recently specified many Performance Metrics related to "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611], which establishes a framework to allow new information to be conveyed in RTCP, supplementing the original report blocks defined in "RTP: A Transport Protocol for Real-Time Applications", [RFC3550].

The "Benchmarking Methodology" WG (BMWG) defined many Performance Metrics for use in laboratory benchmarking of inter-networking technologies.

The "IP Flow Information eXport" (IPFIX) concluded WG specified an IANA process for new Information Elements. Some Performance Metrics related Information Elements are proposed on regular basis.

The "Performance Metrics for Other Layers" (PMOL) a concluded WG, defined some Performance Metrics related to Session Initiation Protocol (SIP) voice quality [RFC6035].

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

Despite the importance of Performance Metrics, there are two related problems for the industry. First, ensuring that when one party requests another party to measure (or report or in some way act on) a particular Performance Metric, then both parties have exactly the same understanding of what Performance Metric is being referred to. Second, discovering which Performance Metrics have been specified, to avoid developing a new Performance Metric that is very similar, but not quite inter-operable. These problems can be addressed by creating a registry of performance metrics. The usual way in which the IETF organizes registries is with Internet Assigned Numbers Authority (IANA), and there is currently no Performance Metrics Registry maintained by the IANA.

This document requests that IANA create and maintain a Performance Metrics Registry, according to the maintenance procedures and the Performance Metrics Registry format defined in this memo. The resulting Performance Metrics Registry is for use by the IETF and others. Although the Registry formatting specifications herein are primarily for registry creation by IANA, any other organization that wishes to create a Performance Metrics Registry MAY use the same formatting specifications for their purposes. The authors make no guarantee of the registry format's applicability to any possible set of Performance Metrics envisaged by other organizations, but encourage others to apply it. In the remainder of this document, unless we explicitly say otherwise, we will refer to the IANA-maintained Performance Metrics Registry as simply the Performance Metrics Registry.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Performance Metric: A Performance Metric is a quantitative measure of performance, targeted to an IETF-specified protocol or targeted to an application transported over an IETF-specified protocol. Examples of Performance Metrics are the FTP response time for a

complete file download, the DNS response time to resolve the IP address, a database logging time, etc. This definition is consistent with the definition of metric in [RFC2330] and broader than the definition of performance metric in [RFC6390].

Registered Performance Metric: A Registered Performance Metric is a Performance Metric expressed as an entry in the Performance Metrics Registry, administered by IANA. Such a performance metric has met all the registry review criteria defined in this document in order to be included in the registry.

Performance Metrics Registry: The IANA registry containing Registered Performance Metrics.

Proprietary Registry: A set of metrics that are registered in a proprietary registry, as opposed to Performance Metrics Registry.

Performance Metrics Experts: The Performance Metrics Experts is a group of designated experts [RFC8126] selected by the IESG to validate the Performance Metrics before updating the Performance Metrics Registry. The Performance Metrics Experts work closely with IANA.

Parameter: A Parameter is an input factor defined as a variable in the definition of a Performance Metric. A Parameter is a numerical or other specified factor forming one of a set that defines a metric or sets the conditions of its operation. All Parameters must be known to measure using a metric and interpret the results. There are two types of Parameters: Fixed and Run-time parameters. For the Fixed Parameters, the value of the variable is specified in the Performance Metrics Registry entry and different Fixed Parameter values result in different Registered Performance Metrics. For the Run-time Parameters, the value of the variable is defined when the metric measurement method is executed and a given Registered Performance Metric supports multiple values for the parameter. Although Run-time Parameters do not change the fundamental nature of the Performance Metric's definition, some have substantial influence on the network property being assessed and interpretation of the results.

Note: Consider the case of packet loss in the following two Active Measurement Method cases. The first case is packet loss as background loss where the Run-time Parameter set includes a very sparse Poisson stream, and only characterizes the times when packets were lost. Actual user streams likely see much higher loss at these times, due to tail drop or radio errors. The second case is packet loss as inverse of throughput where the Run-time Parameter set includes a very dense, bursty

stream, and characterizes the loss experienced by a stream that approximates a user stream. These are both "loss metrics", but the difference in interpretation of the results is highly dependent on the Run-time Parameters (at least), to the extreme where we are actually using loss to infer its compliment: delivered throughput.

Active Measurement Method: Methods of Measurement conducted on traffic which serves only the purpose of measurement and is generated for that reason alone, and whose traffic characteristics are known a priori. The complete definition of Active Methods is specified in section 3.4 of [RFC7799]. Examples of Active Measurement Methods are the measurement methods for the One way delay metric defined in [RFC7679] and the one for round trip delay defined in [RFC2681].

Passive Measurement Method: Methods of Measurement conducted on network traffic, generated either from the end users or from network elements that would exist regardless whether the measurement was being conducted or not. The complete definition of Passive Methods is specified in section 3.6 of [RFC7799]. One characteristic of Passive Measurement Methods is that sensitive information may be observed, and as a consequence, stored in the measurement system.

Hybrid Measurement Method: Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods, to assess Active Metrics, Passive Metrics, or new metrics derived from the a priori knowledge and observations of the stream of interest. The complete definition of Hybrid Methods is specified in section 3.8 of [RFC7799].

3. Scope

This document is intended for two different audiences:

1. For those defining new Registered Performance Metrics, it provides specifications and best practices to be used in deciding which Registered Performance Metrics are useful for a measurement study, instructions for writing the text for each column of the Registered Performance Metrics, and information on the supporting documentation required for the new Performance Metrics Registry entry (up to and including the publication of one or more RFCs or I-Ds describing it).
2. For the appointed Performance Metrics Experts and for IANA personnel administering the new IANA Performance Metrics Registry, it defines a set of acceptance criteria against which

these proposed Registered Performance Metrics should be evaluated.

In addition, this document may be useful for other organizations who are defining a Performance Metric registry of their own, and may re-use the features of the Performance Metrics Registry defined in this document.

This Performance Metrics Registry is applicable to Performance Metrics issued from Active Measurement, Passive Measurement, and any other form of Performance Metric. This registry is designed to encompass Performance Metrics developed throughout the IETF and especially for the technologies specified in the following working groups: IPPM, XRBLOCK, IPFIX, and BMWG. This document analyzes an prior attempt to set up a Performance Metrics Registry, and the reasons why this design was inadequate [RFC6248]. Finally, this document gives a set of guidelines for requesters and expert reviewers of candidate Registered Performance Metrics.

This document makes no attempt to populate the Performance Metrics Registry with initial entries.

Based on [RFC8126] Section 4.3, this document is processed as Best Current Practice (BCP) [RFC2026].

4. Motivation for a Performance Metrics Registry

In this section, we detail several motivations for the Performance Metrics Registry.

4.1. Interoperability

As any IETF registry, the primary use for a registry is to manage a registry for its use within one or more protocols. In the particular case of the Performance Metrics Registry, there are two types of protocols that will use the Performance Metrics in the Performance Metrics Registry during their operation (by referring to the Index values):

- o Control protocol: This type of protocol used to allow one entity to request another entity to perform a measurement using a specific metric defined by the Performance Metrics Registry. One particular example is the LMAP framework [RFC7594]. Using the LMAP terminology, the Performance Metrics Registry is used in the LMAP Control protocol to allow a Controller to request a measurement task to one or more Measurement Agents. In order to enable this use case, the entries of the Performance Metrics Registry must be sufficiently defined to allow a Measurement Agent

implementation to trigger a specific measurement task upon the reception of a control protocol message. This requirement heavily constrains the type of entries that are acceptable for the Performance Metrics Registry.

- o Report protocol: This type of protocol is used to allow an entity to report measurement results to another entity. By referencing to a specific Performance Metrics Registry, it is possible to properly characterize the measurement result data being reported. Using the LMAP terminology, the Performance Metrics Registry is used in the Report protocol to allow a Measurement Agent to report measurement results to a Collector.

It should be noted that the LMAP framework explicitly allows for using not only the IANA-maintained Performance Metrics Registry but also other registries containing Performance Metrics, either defined by other organizations or private ones. However, others who are creating Registries to be used in the context of an LMAP framework are encouraged to use the Registry format defined in this document, because this makes it easier for developers of LMAP Measurement Agents (MAs) to programmatically use information found in those other Registries' entries.

4.2. Single point of reference for Performance Metrics

A Performance Metrics Registry serves as a single point of reference for Performance Metrics defined in different working groups in the IETF. As we mentioned earlier, there are several WGs that define Performance Metrics in the IETF and it is hard to keep track of all them. This results in multiple definitions of similar Performance Metrics that attempt to measure the same phenomena but in slightly different (and incompatible) ways. Having a registry would allow both the IETF community and external people to have a single list of relevant Performance Metrics defined by the IETF (and others, where appropriate). The single list is also an essential aspect of communication about Performance Metrics, where different entities that request measurements, execute measurements, and report the results can benefit from a common understanding of the referenced Performance Metric.

4.3. Side benefits

There are a couple of side benefits of having such a registry. First, the Performance Metrics Registry could serve as an inventory of useful and used Performance Metrics, that are normally supported by different implementations of measurement agents. Second, the results of measurements using the Performance Metrics should be comparable even if they are performed by different implementations

and in different networks, as the Performance Metric is properly defined. BCP 176 [RFC6576] examines whether the results produced by independent implementations are equivalent in the context of evaluating the completeness and clarity of metric specifications. This BCP defines the standards track advancement testing for (active) IPPM metrics, and the same process will likely suffice to determine whether Registered Performance Metrics are sufficiently well specified to result in comparable (or equivalent) results. Registered Performance Metrics which have undergone such testing SHOULD be noted, with a reference to the test results.

5. Criteria for Performance Metrics Registration

It is neither possible nor desirable to populate the Performance Metrics Registry with all combinations of Parameters of all Performance Metrics. The Registered Performance Metrics should be:

1. interpretable by the user.
2. implementable by the software designer,
3. deployable by network operators,
4. accurate, for interoperability and deployment across vendors,
5. Operationally useful, so that it has significant industry interest and/or has seen deployment,
6. Sufficiently tightly defined, so that different values for the Run-time Parameters does not change the fundamental nature of the measurement, nor change the practicality of its implementation.

In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose.

6. Performance Metric Registry: Prior attempt

There was a previous attempt to define a metric registry RFC 4148 [RFC4148]. However, it was obsoleted by RFC 6248 [RFC6248] because it was "found to be insufficiently detailed to uniquely identify IPPM metrics... [there was too much] variability possible when characterizing a metric exactly" which led to the RFC4148 registry having "very few users, if any".

A couple of interesting additional quotes from RFC 6248 might help understand the issues related to that registry.

1. "It is not believed to be feasible or even useful to register every possible combination of Type P, metric parameters, and Stream parameters using the current structure of the IPPM Metrics Registry."
2. "The registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics."
3. "Despite apparent efforts to find current or even future users, no one responded to the call for interest in the RFC 4148 registry during the second half of 2010."

The current approach learns from this by tightly defining each Registered Performance Metric with only a few variable (Run-time) Parameters to be specified by the measurement designer, if any. The idea is that entries in the Performance Metrics Registry stem from different measurement methods which require input (Run-time) parameters to set factors like source and destination addresses (which do not change the fundamental nature of the measurement). The downside of this approach is that it could result in a large number of entries in the Performance Metrics Registry. There is agreement that less is more in this context - it is better to have a reduced set of useful metrics rather than a large set of metrics, some with questionable usefulness.

6.1. Why this Attempt Will Succeed

As mentioned in the previous section, one of the main issues with the previous registry was that the metrics contained in the registry were too generic to be useful. This document specifies stricter criteria for performance metric registration (see section 6), and imposes a group of Performance Metrics Experts that will provide guidelines to assess if a Performance Metric is properly specified.

Another key difference between this attempt and the previous one is that in this case there is at least one clear user for the Performance Metrics Registry: the LMAP framework and protocol. Because the LMAP protocol will use the Performance Metrics Registry values in its operation, this actually helps to determine if a metric is properly defined. In particular, since we expect that the LMAP control protocol will enable a controller to request a measurement agent to perform a measurement using a given metric by embedding the Performance Metrics Registry value in the protocol, a metric is properly specified if it is defined well-enough so that it is possible (and practical) to implement the metric in the measurement agent. This was the failure of the previous attempt: a registry entry with an undefined Type-P (section 13 of RFC 2330 [RFC2330]) allows implementation to be ambiguous.

7. Definition of the Performance Metric Registry

This Performance Metrics Registry is applicable to Performance Metrics used for Active Measurement, Passive Measurement, and any other form of Performance Metric. Each category of measurement has unique properties, so some of the columns defined below are not applicable for a given metric category. In this case, the column(s) SHOULD be populated with the "NA" value (Non Applicable). However, the "NA" value MUST NOT be used by any metric in the following columns: Identifier, Name, URI, Status, Requester, Revision, Revision Date, Description. In the future, a new category of metrics could require additional columns, and adding new columns is a recognized form of registry extension. The specification defining the new column(s) MUST give guidelines to populate the new column(s) for existing entries (in general).

The columns of the Performance Metrics Registry are defined below. The columns are grouped into "Categories" to facilitate the use of the registry. Categories are described at the 7.x heading level, and columns are at the 7.x.y heading level. The Figure below illustrates this organization. An entry (row) therefore gives a complete description of a Registered Performance Metric.

Each column serves as a check-list item and helps to avoid omissions during registration and expert review.

Registry Categories and Columns, shown as

Category

 Column | Column |

Summary

 Identifier | Name | URIs | Desc. | Reference | Change Controller | Ver |

Metric Definition

 Reference Definition | Fixed Parameters |

Method of Measurement

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

Output

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

Administrative Information

 Status | Request | Rev | Rev.Date |

Comments and Remarks

7.1. Summary Category

7.1.1. Identifier

A numeric identifier for the Registered Performance Metric. This identifier MUST be unique within the Performance Metrics Registry.

The Registered Performance Metric unique identifier is an unbounded integer (range 0 to infinity).

The Identifier 0 should be Reserved. The Identifier values from 64512 to 65536 are reserved for private use.

When adding newly Registered Performance Metrics to the Performance Metrics Registry, IANA SHOULD assign the lowest available identifier to the new Registered Performance Metric.

If a Performance Metrics Expert providing review determines that there is a reason to assign a specific numeric identifier, possibly leaving a temporary gap in the numbering, then the Performance Expert SHALL inform IANA of this decision.

7.1.2. Name

As the name of a Registered Performance Metric is the first thing a potential human implementor will use when determining whether it is suitable for their measurement study, it is important to be as precise and descriptive as possible. In future, users will review the names to determine if the metric they want to measure has already been registered, or if a similar entry is available as a basis for creating a new entry.

Names are composed of the following elements, separated by an underscore character "_":

MetricType_Method_SubTypeMethod_... Spec_Units_Output

- o MetricType: a combination of the directional properties and the metric measured, such as:

RTDelay (Round Trip Delay)

RTDNS (Response Time Domain Name Service)

RLDNS (Response Loss Domain Name Service)

OWDelay (One Way Delay)

RTLoss (Round Trip Loss)

OWLoss (One Way Loss)

OWPDV (One Way Packet Delay Variation)

OWIPDV (One Way Inter-Packet Delay Variation)

OWReorder (One Way Packet Reordering)

OWDuplic (One Way Packet Duplication)

OWBTC (One Way Bulk Transport Capacity)

OWMBM (One Way Model Based Metric)

SPMonitor (Single Point Monitor)

MPMonitor (Multi-Point Monitor)

- o Method: One of the methods defined in [RFC7799], such as:

Active (depends on a dedicated measurement packet stream and observations of the stream)

Passive (depends **solely** on observation of one or more existing packet streams)

HybridType1 (observations on one stream that combine both active and passive methods)

HybridType2 (observations on two or more streams that combine both active and passive methods)

Spatial (Spatial Metric of RFC5644)

- o SubTypeMethod: One or more sub-types to further describe the features of the entry, such as:

ICMP (Internet Control Message Protocol)

IP (Internet Protocol)

DSCPxx (where xx is replaced by a Diffserv code point)

UDP (User Datagram Protocol)

TCP (Transport Control Protocol)

QUIC (QUIC transport protocol)

HS (Hand-Shake, such as TCP's 3-way HS)

Poisson (Packet generation using Poisson distribution)

Periodic (Periodic packet generation)

SendOnRcv (Sender keeps one packet in-transit by sending when previous packet arrives)

PayloadxxxxB (where xxxx is replaced by an integer, the number of octets in the Payload)

SustainedBurst (Capacity test, worst case)

StandingQueue (test of bottleneck queue behavior)

SubTypeMethod values are separated by a hyphen "-" character, which indicates that they belong to this element, and that their order is unimportant when considering name uniqueness.

- o Spec: RFC number and major section number that specifies this Registry entry in the form RFCXXXXsecY, such as RFC7799sec3. Note: the RFC number is not the Primary Reference specification for the metric definition, such as [RFC7679] for One-way Delay; it will contain the placeholder "RFCXXXXsecY" until the RFC number is assigned to the specifying document, and would remain blank in private registry entries without a corresponding RFC.
- o Units: The units of measurement for the output, such as:
 - Seconds
 - Ratio (unitless)
 - Percent (value multiplied by 100)
 - Logical (1 or 0)
 - Packets
 - BPS (Bits per Second)
 - PPS (Packets per Second)
 - EventTotal (for unit-less counts)
 - Multiple (more than one type of unit)
 - Enumerated (a list of outcomes)
 - Unitless
- o Output: The type of output resulting from measurement, such as:
 - Singleton
 - Raw (multiple Singletons)
 - Count
 - Minimum
 - Maximum

Median

Mean

95Percentile (95th Percentile)

99Percentile (99th Percentile)

StdDev (Standard Deviation)

Variance

PFI (Pass, Fail, Inconclusive)

FlowRecords (descriptions of flows observed)

LossRatio (lost packets to total packets, <=1)

An example is:

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile

as described in section 4 of [I-D.ietf-ippm-initial-registry].

Note that private registries following the format described here SHOULD use the prefix "Priv_" on any name to avoid unintended conflicts (further considerations are described in section 10). Private registry entries usually have no specifying RFC, thus the Spec: element has no clear interpretation.

7.1.3. URIs

The URIs column MUST contain a URL [RFC3986] that uniquely identifies and locates the metric entry so it is accessible through the Internet. The URL points to a file containing all the human-readable information for one registry entry. The URL SHALL reference a target file that is HTML-formated and contains URLs to referenced sections of HTML-ized RFCs. These target files for different entries can be more easily edited and re-used when preparing new entries. The exact form of the URL for each target file will be determined by IANA and reside on "iana.org". The major sections of [I-D.ietf-ippm-initial-registry] provide an example of a target file in HTML form (sections 4 and higher).

7.1.4. Description

A Registered Performance Metric description is a written representation of a particular Performance Metrics Registry entry. It supplements the Registered Performance Metric name to help Performance Metrics Registry users select relevant Registered Performance Metrics.

7.1.5. Reference

This entry gives the specification containing the candidate registry entry which was reviewed and agreed, if such an RFC or other specification exists.

7.1.6. Change Controller

This entry names the entity responsible for approving revisions to the registry entry, and SHALL provide contact information (for an individual, where appropriate).

7.1.7. Version (of Registry Format)

This entry gives the version number for the registry format used. Formats complying with this memo MUST use 1.0. The version number SHALL NOT change unless a new RFC is published that changes the registry format.

7.2. Metric Definition Category

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

7.2.1. Reference Definition

This entry provides a reference (or references) to the relevant section(s) of the document(s) that define the metric, as well as any supplemental information needed to ensure an unambiguous definition for implementations. The reference needs to be an immutable document, such as an RFC; for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification.

7.2.2. Fixed Parameters

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

Where referenced metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Fixed Parameters. As an example for active metrics, Fixed Parameters determine most or all of the IPPM Framework convention "packets of Type-P" as described in [RFC2330], such as transport protocol, payload length, TTL, etc. An example for passive metrics is for RTP packet loss calculation that relies on the validation of a packet as RTP which is a multi-packet validation controlled by MIN_SEQUENTIAL as defined by [RFC3550]. Varying MIN_SEQUENTIAL values can alter the loss report and this value could be set as a Fixed Parameter.

Parameters MUST have well-defined names. For human readers, the hanging indent style is preferred, and any Parameter names and definitions that do not appear in the Reference Method Specification MUST appear in this column (or Run-time Parameters column).

Parameters MUST have a well-specified data format.

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

7.3. Method of Measurement Category

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

7.3.1. Reference Method

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

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

7.3.2. Packet Stream Generation

This column applies to Performance Metrics that generate traffic as part of their Measurement Method, including but not necessarily limited to Active metrics. The generated traffic is referred as a stream and this column describes its characteristics.

Each entry for this column contains the following information:

- o Value: The name of the packet stream scheduling discipline
- o Reference: the specification where the parameters of the stream are defined

The packet generation stream may require parameters such as the average packet rate and distribution truncation value for streams with Poisson-distributed inter-packet sending times. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

The simplest example of stream specification is Singleton scheduling (see [RFC2330]), where a single atomic measurement is conducted. Each atomic measurement could consist of sending a single packet (such as a DNS request) or sending several packets (for example, to request a webpage). Other streams support a series of atomic measurements in a "sample", with a schedule defining the timing between each transmitted packet and subsequent measurement. Principally, two different streams are used in IPPM metrics, Poisson distributed as described in [RFC2330] and Periodic as described in [RFC3432]. Both Poisson and Periodic have their own unique parameters, and the relevant set of parameters names and values should be included either in the Fixed Parameters column or in the Run-time parameter column.

7.3.3. Traffic Filter

This column applies to Performance Metrics that observe packets flowing through (the device with) the measurement agent i.e. that is not necessarily addressed to the measurement agent. This includes but is not limited to Passive Metrics. The filter specifies the traffic that is measured. This includes protocol field values/ranges, such as address ranges, and flow or session identifiers.

The traffic filter itself depends on needs of the metric itself and a balance of an operator's measurement needs and a user's need for privacy. Mechanics for conveying the filter criteria might be the BPF (Berkeley Packet Filter) or PSAMP [RFC5475] Property Match

Filtering which reuses IPFIX [RFC7012]. An example BPF string for matching TCP/80 traffic to remote destination net 192.0.2.0/24 would be "dst net 192.0.2.0/24 and tcp dst port 80". More complex filter engines might be supported by the implementation that might allow for matching using Deep Packet Inspection (DPI) technology.

The traffic filter includes the following information:

Type: the type of traffic filter used, e.g. BPF, PSAMP, OpenFlow rule, etc. as defined by a normative reference

Value: the actual set of rules expressed

7.3.4. Sampling Distribution

The sampling distribution defines out of all the packets that match the traffic filter, which one of those are actually used for the measurement. One possibility is "all" which implies that all packets matching the Traffic filter are considered, but there may be other sampling strategies. It includes the following information:

Value: the name of the sampling distribution

Reference definition: pointer to the specification where the sampling distribution is properly defined.

The sampling distribution may require parameters. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

Sampling and Filtering Techniques for IP Packet Selection are documented in the PSAMP (Packet Sampling) [RFC5475], while the Framework for Packet Selection and Reporting, [RFC5474] provides more background information. The sampling distribution parameters might be expressed in terms of the Information Model for Packet Sampling Exports, [RFC5477], and the Flow Selection Techniques, [RFC7014].

7.3.5. Run-time Parameters

Run-Time Parameters are Parameters 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 Performance Metrics Registry (like the Fixed Parameters), rather these parameters are listed as an aid to the measurement system implementer or user (they must be left as variables, and supplied on execution).

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

Parameters MUST have well defined names. For human readers, the hanging indent style is preferred, and the names and definitions that do not appear in the Reference Method Specification MUST appear in this column.

A Data Format for each Run-time Parameter MUST be specified in this column, to simplify the control and implementation of measurement devices. For example, parameters that include an IPv4 address can be encoded as a 32 bit integer (i.e. binary base64 encoded value) or ip-address as defined in [RFC6991]. The actual encoding(s) used must be explicitly defined for each Run-time parameter. IPv6 addresses and options MUST be accomodated, allowing Registered Metrics to be used in either address family.

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

7.3.6. Role

In some methods of measurement, there may be several roles defined, e.g., for a one-way packet delay active measurement there is one measurement agent that generates the packets and another agent that receives the packets. This column contains the name of the Role(s) for this particular entry. In the one-way delay example above, there should be two entries in the Role registry column, one for each Role (Source and Destination). When a measurement agent is instructed to perform the "Source" Role for one-way delay metric, the agent knows that it is required to generate packets. The values for this field are defined in the reference method of measurement (and this frequently results in abbreviated role names such as "Src").

When the Role column of a registry entry defines more than one Role, then the Role SHALL be treated as a Run-time Parameter and supplied for execution. It should be noted that the LMAP framework [RFC7594] distinguishes the Role from other Run-time Parameters, and defines a special parameter "Roles" inside the registry-grouping function list in the LMAP YANG model[RFC8194].

7.4. Output Category

For entries which involve a stream and many singleton measurements, a statistic may be specified in this column to summarize the results to

a single value. If the complete set of measured singletons is output, this will be specified here.

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

7.4.1. Type

This column contains the name of the output type. The output type defines a single type of result that the metric produces. It can be the raw results (packet send times and singleton metrics), or it can be a summary statistic. The specification of the output type MUST define the format of the output. In some systems, format specifications will simplify both measurement implementation and collection/storage tasks. Note that if two different statistics are required from a single measurement (for example, both "Xth percentile mean" and "Raw"), then a new output type must be defined ("Xth percentile mean AND Raw"). See the Naming section above for a list of Output Types.

7.4.2. Reference Definition

This column contains a pointer to the specification(s) where the output type and format are defined.

7.4.3. Metric Units

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

When a sample of singletons (see Section 11 of [RFC2330] for definitions of these terms) is collected, this entry will specify the units for each measured value.

7.4.4. Calibration

Some specifications for Methods of Measurement include the possibility to perform an error calibration. Section 3.7.3 of [RFC7679] is one example. In the registry entry, this field will identify a method of calibration for the metric, and when available, the measurement system SHOULD perform the calibration when requested and produce the output with an indication that it is the result of a calibration method. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

7.5. Administrative information

7.5.1. Status

The status of the specification of this Registered Performance Metric. Allowed values are 'current' and 'deprecated'. All newly defined Information Elements have 'current' status.

7.5.2. Requester

The requester for the Registered Performance Metric. The requester MAY be a document, such as RFC, or person.

7.5.3. Revision

The revision number of a Registered Performance Metric, starting at 0 for Registered Performance Metrics at time of definition and incremented by one for each revision.

7.5.4. Revision Date

The date of acceptance or the most recent revision for the Registered Performance Metric. The date SHALL be determined by the reviewing Performance Metrics Expert in the case of Expert Review, or by IANA in the case of Standards Action.

7.6. Comments and Remarks

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

8. The Life-Cycle of Registered Performance Metrics

Once a Performance Metric or set of Performance Metrics has been identified for a given application, candidate Performance Metrics Registry entry specifications prepared in accordance with Section 7 should be submitted to IANA to follow the process for review by the Performance Metric Experts, as defined below. This process is also used for other changes to the Performance Metrics Registry, such as deprecation or revision, as described later in this section.

It is desirable that the author(s) of a candidate Performance Metrics Registry entry seek review in the relevant IETF working group, or offer the opportunity for review on the working group mailing list.

8.1. Adding new Performance Metrics to the Performance Metrics Registry

Requests to add Registered Performance Metrics in the Performance Metrics Registry SHALL be submitted to IANA, which forwards the request to a designated group of experts (Performance Metric Experts) appointed by the IESG; these are the reviewers called for by the Expert Review [RFC8126] policy defined for the Performance Metrics Registry. The Performance Metric Experts review the request for such things as compliance with this document, compliance with other applicable Performance Metric-related RFCs, and consistency with the currently defined set of Registered Performance Metrics.

Submission to IANA MAY be during IESG review (leading to IETF Standards Action), where an Internet Draft proposes one or more Registered Performance Metrics to be added to the Performance Metrics Registry, including the text of the proposed Registered Performance Metric(s).

Authors of proposed Registered Performance Metrics SHOULD review compliance with the specifications in this document to check their submissions before sending them to IANA.

At least one Performance Metric Expert should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the Performance Metric Experts signify their approval to IANA, and IANA updates the Performance Metrics Registry. If the request is not acceptable, the Performance Metric Experts MAY coordinate with the requester to change the request to be compliant, otherwise IANA SHALL coordinate resolution of issues on behalf of the expert. The Performance Metric Experts MAY choose to reject clearly frivolous or inappropriate change requests outright, but such exceptional circumstances should be rare.

This process should not in any way be construed as allowing the Performance Metric Experts to overrule IETF consensus. Specifically, any Registered Performance Metrics that were added to the Performance Metrics Registry with IETF consensus require IETF consensus for revision or deprecation.

Decisions by the Performance Metric Experts may be appealed as in Section 7 of [RFC8126].

8.2. Revising Registered Performance Metrics

A request for Revision is only permitted when the requested changes maintain backward-compatibility with implementations of the prior Performance Metrics Registry entry describing a Registered Performance Metric (entries with lower revision numbers, but the same Identifier and Name).

The purpose of the Status field in the Performance Metrics Registry is to indicate whether the entry for a Registered Performance Metric is 'current' or 'deprecated'.

In addition, no policy is defined for revising the Performance Metric entries in the IANA Registry or addressing errors therein. To be clear, changes and deprecations within the Performance Metrics Registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, the provisions of this section address the need for revisions.

Revisions are initiated by sending a candidate Registered Performance Metric definition to IANA, as in Section 8.1, identifying the existing Performance Metrics Registry entry, and explaining how and why the existing entry should be revised.

The primary requirement in the definition of procedures for managing changes to existing Registered Performance Metrics is avoidance of measurement interoperability problems; the Performance Metric Experts must work to maintain interoperability above all else. Changes to Registered Performance Metrics may only be done in an interoperable way; necessary changes that cannot be done in a way to allow interoperability with unchanged implementations MUST result in the creation of a new Registered Performance Metric (with a new Name, replacing the RFCXXXsecY portion of the name) and possibly the deprecation of the earlier metric.

A change to a Registered Performance Metric SHALL be determined to be backward-compatible only when:

1. it involves the correction of an error that is obviously only editorial; or
2. it corrects an ambiguity in the Registered Performance Metric's definition, which itself leads to issues severe enough to prevent the Registered Performance Metric's usage as originally defined; or
3. it corrects missing information in the metric definition without changing its meaning (e.g., the explicit definition of 'quantity' semantics for numeric fields without a Data Type Semantics value); or
4. it harmonizes with an external reference that was itself corrected.

If a Performance Metric revision is deemed permissible and backward-compatible by the Performance Metric Experts, according to the rules in this document, IANA SHOULD execute the change(s) in the Performance Metrics Registry. The requester of the change is appended to the original requester in the Performance Metrics Registry. The Name of the revised Registered Performance Metric, including the RFCXXXsecY portion of the name, SHALL remain unchanged (even when the change is the result of IETF Standards Action; the revised registry entry SHOULD reference the new RFC in an appropriate category and column).

Each Registered Performance Metric in the Performance Metrics Registry has a revision number, starting at zero. Each change to a Registered Performance Metric following this process increments the revision number by one.

When a revised Registered Performance Metric is accepted into the Performance Metrics Registry, the date of acceptance of the most recent revision is placed into the revision Date column of the registry for that Registered Performance Metric.

Where applicable, additions to Registered Performance Metrics in the form of text Comments or Remarks should include the date, but such additions may not constitute a revision according to this process.

Older version(s) of the updated metric entries are kept in the registry for archival purposes. The older entries are kept with all fields unmodified (version, revision date) except for the status field that SHALL be changed to "Deprecated".

8.3. Deprecating Registered Performance Metrics

Changes that are not permissible by the above criteria for Registered Performance Metric's revision may only be handled by deprecation. A Registered Performance Metric MAY be deprecated and replaced when:

1. the Registered Performance Metric definition has an error or shortcoming that cannot be permissibly changed as in Section 8.2 Revising Registered Performance Metrics; or
2. the deprecation harmonizes with an external reference that was itself deprecated through that reference's accepted deprecation method.

A request for deprecation is sent to IANA, which passes it to the Performance Metric Experts for review. When deprecating an Performance Metric, the Performance Metric description in the Performance Metrics Registry must be updated to explain the deprecation, as well as to refer to any new Performance Metrics created to replace the deprecated Performance Metric.

The revision number of a Registered Performance Metric is incremented upon deprecation, and the revision Date updated, as with any revision.

The use of deprecated Registered Performance Metrics should result in a log entry or human-readable warning by the respective application.

Names and Metric IDs of deprecated Registered Performance Metrics must not be reused.

The deprecated entries are kept with all fields unmodified, except the version, revision date, and the status field (changed to "Deprecated").

9. Security considerations

This draft defines a registry structure, and does not itself introduce any new security considerations for the Internet. The definition of Performance Metrics for this registry may introduce some security concerns, but the mandatory references should have their own considerations for security, and such definitions should be reviewed with security in mind if the security considerations are not covered by one or more reference standards.

10. IANA Considerations

With the background and processes described in earlier sections, this document requests the following IANA Actions. Note that mock-ups of the implementation of this set of requests have been prepared with IANA's help during development of this memo, and have been captured in the Proceedings of IPPM working group sessions.

10.1. Registry Group

The new registry group SHALL be named, "PERFORMANCE METRICS Group".

10.2. Performance Metric Name Elements

This document specifies the procedure for Performance Metrics Name Element Registry setup. IANA is requested to create a new set of registries for Performance Metric Name Elements called "Registered Performance Metric Name Elements". Each Registry, whose names are listed below:

MetricType:

Method:

SubTypeMethod:

Spec:

Units:

Output:

will contain the current set of possibilities for Performance Metrics Registry Entry Names.

To populate the Registered Performance Metric Name Elements at creation, the IANA is asked to use the lists of values for each name element listed in Section 7.1.2. The Name Elements in each registry are case-sensitive.

When preparing a Metric entry for Registration, the developer SHOULD choose Name elements from among the registered elements. However, if the proposed metric is unique in a significant way, it may be necessary to propose a new Name element to properly describe the metric, as described below.

A candidate Metric Entry RFC or document for Expert Review would propose one or more new element values required to describe the

unique entry, and the new name element(s) would be reviewed along with the metric entry. New assignments for Registered Performance Metric Name Elements will be administered by IANA through Expert Review [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors.

10.3. New Performance Metrics Registry

This document specifies the procedure for Performance Metrics Registry setup. IANA is requested to create a new registry for Performance Metrics called "Performance Metrics Registry". This Registry will contain the following Summary columns:

Identifier:

Name:

URIs:

Description:

Reference:

Change Controller:

Version:

Descriptions of these columns and additional information found in the template for registry entries (categories and columns) are further defined in section Section 7.

The "Identifier" 0 should be Reserved. "The Identifier" values from 64512 to 65536 are reserved for private use.

Names starting with the prefix Priv_ are reserved for private use, and are not considered for registration. The "Name" column entries are further defined in section Section 7.

The "URIs" column will have a URL to the full template of each registry entry. The Registry Entry text SHALL be HTML-ized to aid the reader, with links to reference RFCs (similar to the way that Internet Drafts are HTML-ized, the same tool can perform the function).

The "Reference" column will include an RFC number, an approved specification designator from another standards body, or the contact person.

New assignments for Performance Metrics Registry will be administered by IANA through Expert Review [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors, or by Standards Action. The experts can be initially drawn from the Working Group Chairs, document editors, and members of the Performance Metrics Directorate, among other sources of experts.

Extensions of the Performance Metrics Registry require IETF Standards Action. Only one form of registry extension is envisaged:

1. Adding columns, or both categories and columns, to accommodate unanticipated aspects of new measurements and metric categories.

If the Performance Metrics Registry is extended in this way, the Version number of future entries complying with the extension SHALL be incremented (either in the unit or tenths digit, depending on the degree of extension).

11. Acknowledgments

Thanks to Brian Trammell and Bill Cerveny, IPPM chairs, for leading some brainstorming sessions on this topic. Thanks to Barbara Stark and Juergen Schoenwaelder for the detailed feedback and suggestions. Thanks to Andrew McGregor for suggestions on metric naming. Thanks to Michelle Cotton for her early IANA review, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL.

12. References

12.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://www.rfc-editor.org/info/rfc2330>>.

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [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, <<https://www.rfc-editor.org/info/rfc6576>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informative References

- [I-D.ietf-ippm-initial-registry]
Morton, A., Bagnulo, M., Eardley, P., and K. D'Souza, "Initial Performance Metrics Registry Entries", draft-ietf-ippm-initial-registry-11 (work in progress), March 2019.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, DOI 10.17487/RFC3432, November 2002, <<https://www.rfc-editor.org/info/rfc3432>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.

- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<https://www.rfc-editor.org/info/rfc3611>>.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, DOI 10.17487/RFC4148, August 2005, <<https://www.rfc-editor.org/info/rfc4148>>.
- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <<https://www.rfc-editor.org/info/rfc5475>>.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, DOI 10.17487/RFC5477, March 2009, <<https://www.rfc-editor.org/info/rfc5477>>.
- [RFC6035] Pendleton, A., Clark, A., Johnston, A., and H. Sinnreich, "Session Initiation Protocol Event Package for Voice Quality Reporting", RFC 6035, DOI 10.17487/RFC6035, November 2010, <<https://www.rfc-editor.org/info/rfc6035>>.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, DOI 10.17487/RFC6248, April 2011, <<https://www.rfc-editor.org/info/rfc6248>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<https://www.rfc-editor.org/info/rfc7012>>.
- [RFC7014] D'Antonio, S., Zseby, T., Henke, C., and L. Peluso, "Flow Selection Techniques", RFC 7014, DOI 10.17487/RFC7014, September 2013, <<https://www.rfc-editor.org/info/rfc7014>>.

- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <<https://www.rfc-editor.org/info/rfc8194>>.

Authors' Addresses

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

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

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

Email: bclaise@cisco.com

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: philip.eardley@bt.com

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

Email: acmorton@att.com

Aamer Akhter
Consultant
118 Timber Hitch
Cary, NC
USA

Email: aakhter@gmail.com

IPPM WG
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

R. Civil
Ciena Corporation
A. Morton
AT&T Labs
R. Rahman
Cisco Systems
M. Jethanandani
Xoriant Corporation
K. Pentikousis, Ed.
Travelping
July 2, 2018

Two-Way Active Measurement Protocol (TWAMP) Data Model
draft-ietf-ippm-twamp-yang-13

Abstract

This document specifies a data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP). The document defines the TWAMP data model through Unified Modeling Language (UML) class diagrams and formally specifies it using a NDMA-compliant YANG model.

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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 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

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

Table of Contents

1. Introduction	3
1.1. Motivation	3
1.2. Terminology	3
1.3. Document Organization	4
2. Scope, Model, and Applicability	4
3. Data Model Overview	5
3.1. Control-Client	6
3.2. Server	7
3.3. Session-Sender	7
3.4. Session-Reflector	8
4. Data Model Parameters	8
4.1. Control-Client	8
4.2. Server	11
4.3. Session-Sender	13
4.4. Session-Reflector	14
5. Data Model	16
5.1. YANG Tree Diagram	16
5.2. YANG Module	19
6. Data Model Examples	48
6.1. Control-Client	48
6.2. Server	50
6.3. Session-Sender	51
6.4. Session-Reflector	52
7. Security Considerations	55
8. IANA Considerations	56
9. Acknowledgements	57
10. Contributors	57
11. References	57
11.1. Normative References	57
11.2. Informative References	59
Appendix A. Detailed Data Model Examples	60
A.1. Control-Client	60
A.2. Server	62
A.3. Session-Sender	64
A.4. Session-Reflector	65
Appendix B. TWAMP Operational Commands	67
Authors' Addresses	67

1. Introduction

The Two-Way Active Measurement Protocol (TWAMP) [RFC5357] is used to measure network performance parameters such as latency, bandwidth, and packet loss by sending probe packets and measuring their experience in the network. To date, TWAMP implementations do not come with a standard management framework, and, as such, implementers have no choice except to provide a proprietary mechanism. This document addresses this gap by defining the model using UML [UML] class diagrams, and formally specifying a NMDA-complaint [RFC8342] TWAMP data model using YANG 1.1 [RFC7950].

1.1. Motivation

In current TWAMP deployments the lack of a standardized data model limits the flexibility to dynamically instantiate TWAMP-based measurements across equipment from different vendors. In large, virtualized, and dynamically instantiated infrastructures where network functions are placed according to orchestration algorithms, proprietary mechanisms for managing TWAMP measurements pose severe limitations with respect to programmability.

Two major trends call for standardizing TWAMP management aspects. First, it is expected that in the coming years large-scale and multi-vendor TWAMP deployments will become the norm. From an operations perspective, using several vendor-specific TWAMP configuration mechanisms when one standard mechanism could provide an alternative is expensive and inefficient. Second, the increasingly software-defined and virtualized nature of network infrastructures, based on dynamic service chains [NSC] and programmable control and management planes Software-Defined Networking (SDN): Layers and Architecture Terminology [RFC7426] requires a well-defined data model for TWAMP implementations. This document defines such a TWAMP data model and specifies it formally using the YANG 1.1 [RFC7950] data modeling language.

Note to RFC Editor:

Please replace the date 2018-07-02 in Section 5.2 of the draft with the date of publication of this draft as a RFC. Also, replace reference to RFC XXXX, and draft-ietf-ippm-port-twamp-test with the RFC numbers assigned to the drafts.

1.2. Terminology

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

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Document Organization

The rest of this document is organized as follows. Section 2 presents the scope and applicability of this document. Section 3 provides a high-level overview of the TWAMP data model. Section 4 details the configuration parameters of the data model and Section 5 specifies in YANG the TWAMP data model. Section 6 lists illustrative examples which conform to the YANG data model specified in this document. Appendix A elaborates these examples further.

2. Scope, Model, and Applicability

The purpose of this document is the specification of a vendor-independent data model for TWAMP implementations.

Figure 1 illustrates a redrawn version of the TWAMP logical model found in Section 1.2 of TWAMP [RFC5357]. The figure is annotated with pointers to the UML [UML] diagrams provided in this document and associated with the data model of the four logical entities in a TWAMP deployment, namely the TWAMP Control-Client, Server, Session-Sender and Session-Reflector. A UML [UML] Notation Guide is available in Section 5 of the said document.

As per TWAMP [RFC5357], unlabeled links in Figure 1 are left unspecified and may be proprietary protocols.

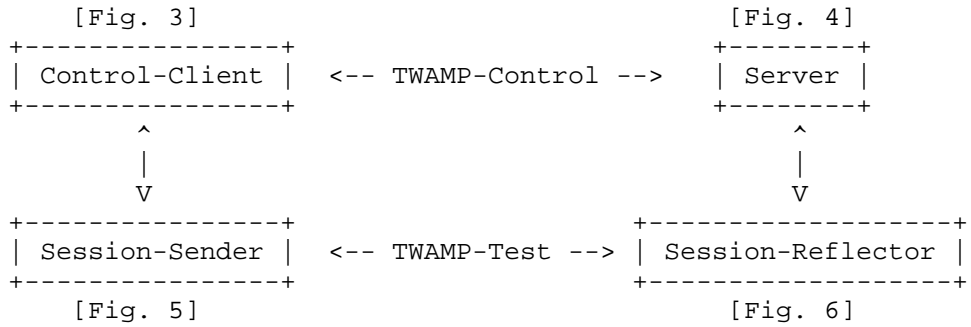


Figure 1: Annotated TWAMP logical model

As per TWAMP [RFC5357], a TWAMP implementation may follow a simplified logical model, in which the same node acts both as Control-Client and Session-Sender, while another node acts at the same time as TWAMP Server and Session-Reflector. Figure 2 illustrates this simplified logical model and indicates the

interaction between the TWAMP configuration client and server using, for instance, NETCONF [RFC6241] or RESTCONF [RFC8040].

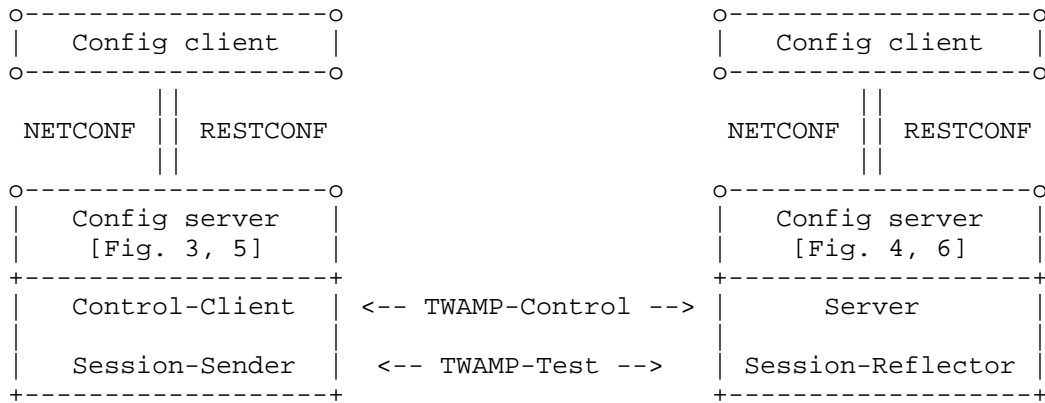


Figure 2: Simplified TWAMP model and protocols

The data model defined in this document is orthogonal to the specific protocol used between the Config client and Config server to communicate the TWAMP configuration parameters.

Operational actions such as how TWAMP-Test sessions are started and stopped, how performance measurement results are retrieved, or how stored results are cleared, and so on, are not addressed by the configuration model defined in this document. As noted above, such operational actions are not part of the TWAMP specification TWAMP [RFC5357] and hence are out of scope of this document. See also Appendix B. In addition, for operational state, current work in Registry for Performance Metrics [I-D.ietf-ippm-metric-registry], can be used to develop an independent model for the performance metrics that need to be captured and retrieved.

3. Data Model Overview

The TWAMP data model includes four categories of configuration items.

First, global configuration items relate to parameters that are set on a per device level. For example, the administrative status of the device with respect to whether it allows TWAMP sessions and, if so, in what capacity (e.g. Control-Client, Server or both), is a typical instance of a global configuration item.

A second category includes attributes that can be configured on a per TWAMP-Control connection basis, such as the Server IP address.

A third category includes attributes related to per TWAMP-Test session attributes, for instance setting different values in the Differentiated Services Code Point (DSCP) field.

Finally, the data model includes attributes that relate to the operational state of the TWAMP implementation.

As the TWAMP data model is described in the remaining sections of this document, readers should keep in mind the functional entity grouping illustrated in Figure 1.

3.1. Control-Client

A TWAMP Control-Client has an administrative status field set at the device level that indicates whether the node is enabled to function as such.

Each TWAMP Control-Client is associated with zero or more TWAMP-Control connections. The main configuration parameters of each control connection are:

- o A name which can be used to uniquely identify at the Control-Client a particular control connection. This name is necessary for programmability reasons because at the time of creation of a TWAMP-Control connection not all IP and TCP port number information needed to uniquely identify the connection is available.
- o The IP address of the interface the Control-Client will use for connections.
- o The IP address of the remote TWAMP Server.
- o Authentication and encryption attributes such as KeyID, Token and the Client Initialization Vector (Client-IV); see also Section 3.1 in OWAMP [RFC4656] and Randomness Requirements for Security [RFC4086].

Each TWAMP-Control connection, in turn, is associated with zero or more TWAMP-Test sessions. For each test session, the following configuration items should be noted:

- o The test session name uniquely identifies a particular test session at the Control-Client and Session-Sender. Similar to the control connections above, this unique test session name is needed because at the time of creation of a TWAMP-Test session, for example, the source UDP port number is not known to uniquely identify the test session.

- o The IP address and UDP port number of the Session-Sender on the path under test by TWAMP.
- o The IP address and UDP port number of the Session-Reflector on said path.
- o Information pertaining to the test packet stream, such as the test starting time, which performance metric is to be used, as defined in Registry for Performance Metrics [I-D.ietf-ippm-metric-registry], or whether the test should be repeated.

3.2. Server

Each TWAMP Server has an administrative status field set at the device level to indicate whether the node is enabled to function as a TWAMP Server.

Each Server is associated with zero or more TWAMP-Control connections. Each control connection is uniquely identified by the 4-tuple {Control-Client IP address, Control-Client TCP port number, Server IP address, Server TCP port}. Control connection configuration items on a TWAMP Server are read-only.

3.3. Session-Sender

A TWAMP Session-Sender has an administrative status field set at the device level that indicates whether the node is enabled to function as such.

There is one Session-Sender instance for each TWAMP-Test session that is initiated from the sending device. Primary configuration fields include:

- o The test session name MUST be identical to the corresponding test session name on the TWAMP Control-Client (Section 3.1).
- o The control connection name, which along with the test session name uniquely identify the TWAMP Session-Sender instance.
- o Information pertaining to the test packet stream, such as, the number of test packets and the packet distribution to be employed; see also Network performance measurement with periodic streams [RFC3432].

3.4. Session-Reflector

Each TWAMP Session-Reflector has an administrative status field set at the device level to indicate whether the node is enabled to function as such.

Each Session-Reflector is associated with zero or more TWAMP-Test sessions. For each test session, the REFWAIT timeout parameter, which determines whether to discontinue the session if no packets have been received (TWAMP [RFC5357], Section 4.2), can be configured.

Read-only access to other data model parameters, such as the Sender IP address, is foreseen. Each test session can be uniquely identified by the 4-tuple mentioned in Section 3.2.

4. Data Model Parameters

This section defines the TWAMP data model using UML [UML] and introduces selected parameters associated with the four TWAMP logical entities. The complete TWAMP data model specification is provided in the YANG module presented in Section 5.2.

4.1. Control-Client

The client container (see Figure 3) holds items that are related to the configuration of the TWAMP Control-Client logical entity (recall Figure 1).

The client container includes an administrative configuration parameter (client/admin-state) that indicates whether the device is allowed to initiate TWAMP-Control connections.

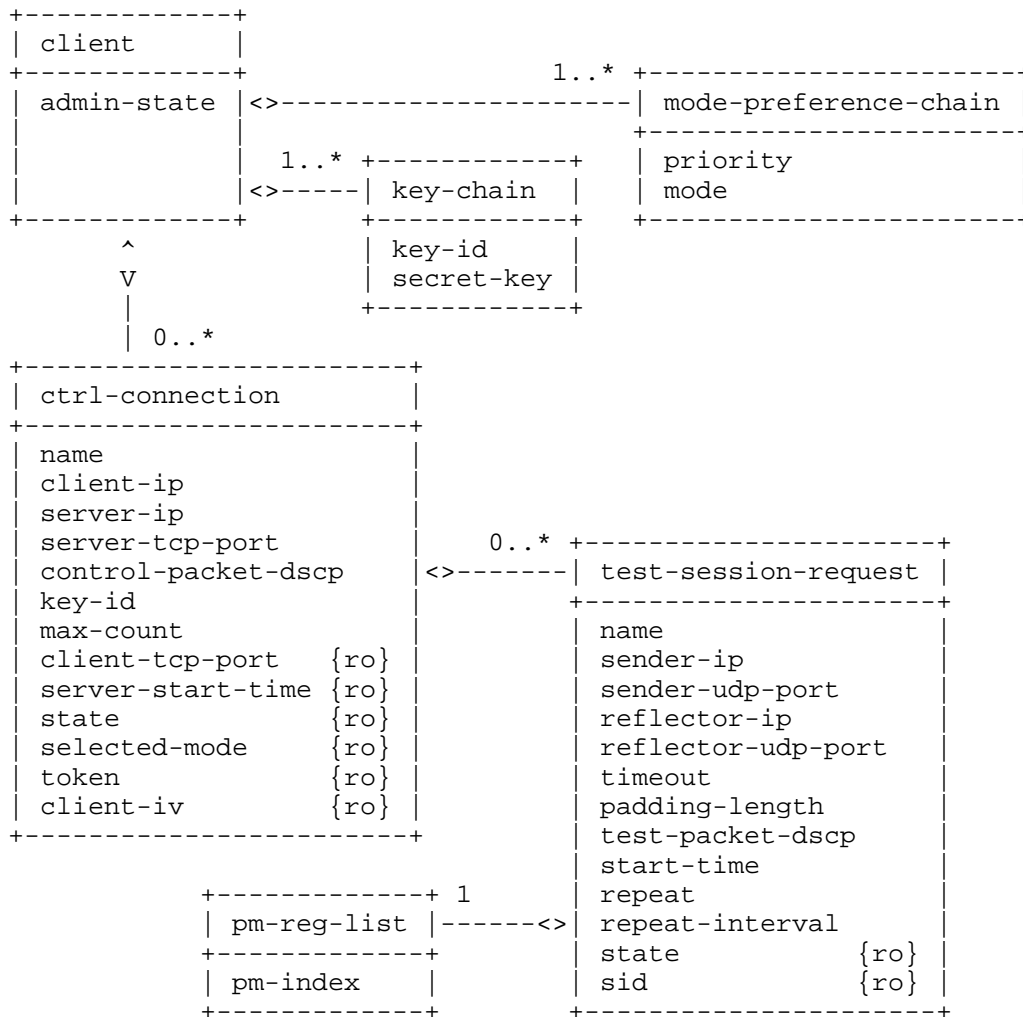


Figure 3: TWAMP Control-Client UML class diagram

The client container holds a list (mode-preference-chain) which specifies the Mode values according to their preferred order of use by the operator of this Control-Client, including the authentication and encryption Modes. Specifically, mode-preference-chain lists the mode and its corresponding priority, as a 16-bit unsigned integer. Values for the priority start with zero, the highest priority, and decreasing priority value is indicated by every increase in value by one.

Depending on the Modes available in the Server Greeting, the Control-Client MUST choose the highest priority Mode from the configured mode-preference-chain list.

Note that the list of preferred Modes may set multiple bit positions independently, such as when referring to the extended TWAMP features in Mixed Security Mode for TWAMP [RFC5618], Individual Session Control Feature for TWAMP [RFC5938], TWAMP Reflect Octets and Symmetrical Size Features [RFC6038], and IKEv2-Derived Shared Secret Key for OWAMP and TWAMP [RFC7717]. If the Control-Client cannot determine an acceptable Mode, or when the bit combinations do not make sense, e.g., both authenticated and unauthenticated bit are set, it MUST respond with zero Mode bits set in the Set-up Response message, indicating it will not continue with the control connection.

In addition, the client container holds a list named key-chain which relates key-id with the respective secret-key. Both the Server and the Control-Client use the same mappings from key-id to secret-key (in Figure 3); in order for this to work properly, key-id must be unique across all systems in the administrative domain. The Server, being prepared to conduct sessions with more than one Control-Client, uses key-id to choose the appropriate secret-key; a Control-Client would typically have different secret keys for different Servers. The secret-key is the shared secret, of type binary and the length SHOULD contain at least 128 bits of entropy. The key-id and secret-key encoding SHOULD follow Section 9.8 of YANG [RFC7950]. The derived key length (dkLen in PKCS #5: Password-Based Cryptography Specification Version 2.1 [RFC8018]) MUST be 16 octets for the AES Session-key used for encryption and 32 octets for the HMAC-SHA1 Session-key used for authentication; see also Section 6.10 of OWAMP [RFC4656].

Each client container also holds a list of control connections, where each item in the list describes a TWAMP control connection initiated by this Control-Client. There SHALL be one ctrl-connection per TWAMP-Control (TCP) connection that is to be initiated from this device.

In turn, each ctrl-connection holds a test-session-request list. Each test-session-request holds information associated with the Control-Client for this test session. This includes information associated with the Request-TW-Session/Accept-Session message exchange (see Section 3.5 of TWAMP [RFC5357]).

There SHALL be one instance of test-session-request for each TWAMP-Test session that is to be negotiated by this TWAMP-Control connection via a Request-TW-Session/Accept-Session exchange.

The Control-Client is also responsible for scheduling TWAMP-Test sessions, therefore test-session-request holds information related to these actions (e.g. pm-index, repeat-interval).

4.2. Server

The server container (see Figure 4) holds items that are related to the configuration of the TWAMP Server logical entity (recall Figure 1).

The server container includes an administrative configuration parameter (server/admin-state) that indicates whether the device is allowed to receive TWAMP-Control connections.

A device operating in the Server role cannot configure attributes on a per TWAMP-Control connection basis, as it has no foreknowledge of the incoming TWAMP-Control connections to be received. Consequently, any parameter that the Server might want to apply to an incoming control connection must be configured at the overall Server level and applied to all incoming TWAMP-Control connections.

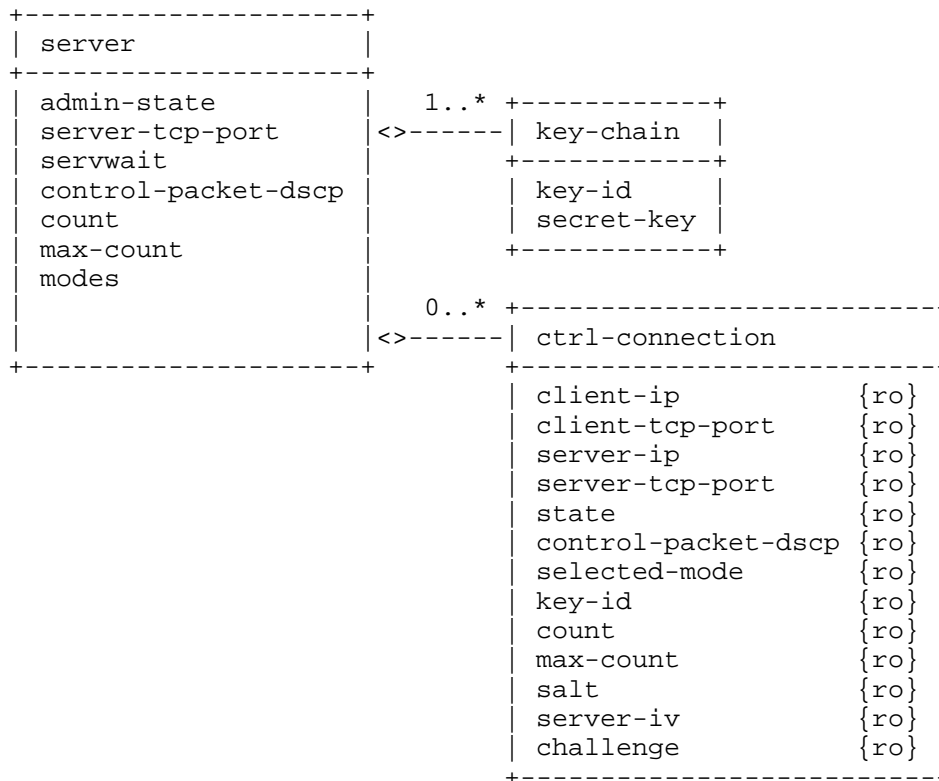


Figure 4: TWAMP Server UML class diagram

Each server container holds a list named key-chain which relates key-id with the respective secret-key. As mentioned in Section 4.1, both the Server and the Control-Client use the same mapping from key-id to shared secret-key; in order for this to work properly, key-id must be unique across all the systems in the administrative domain. The Server, being prepared to conduct sessions with more than one Control-Client, uses key-id to choose the appropriate secret-key; a Control-Client would typically have different secret keys for different Servers. The key-id tells the Server which shared secret-key the Control-Client wishes to use for authentication or encryption.

Each incoming control connection active on the Server is represented by a ctrl-connection. There SHALL be one ctrl-connection per incoming TWAMP-Control (TCP) connection that is received and active on the Server. Each ctrl-connection can be uniquely identified by the 4-tuple {client-ip, client-tcp-port, server-ip, server-tcp-port}. All items in the ctrl-connection list are read-only.

4.3. Session-Sender

The session-sender container, illustrated in Figure 5, holds items that are related to the configuration of the TWAMP Session-Sender logical entity.

The session-sender container includes an administrative parameter (session-sender/admin-state) that controls whether the device is allowed to initiate TWAMP-Test sessions.

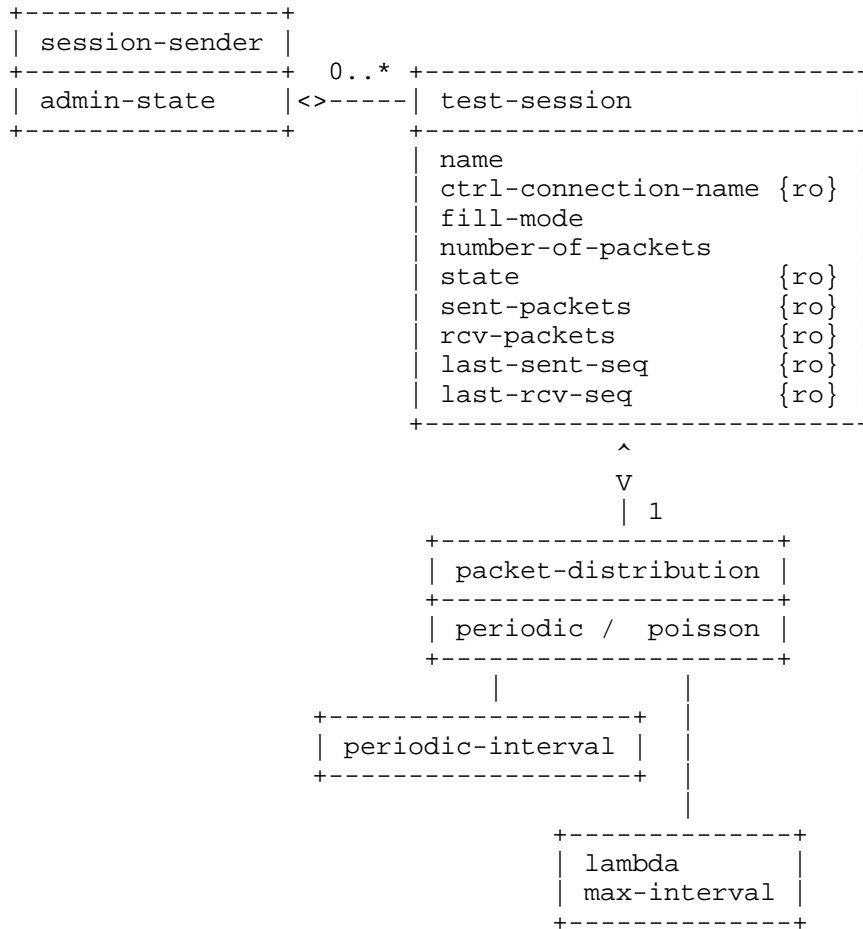


Figure 5: TWAMP Session-Sender UML class diagram

Each TWAMP-Test session initiated by the Session-Sender will be represented by an instance of a test-session object. There SHALL be

one instance of test-session for each TWAMP-Test session for which packets are being sent.

4.4. Session-Reflector

The session-reflector container, illustrated in Figure 6, holds items that are related to the configuration of the TWAMP Session-Reflector logical entity.

The session-reflector container includes an administrative parameter (session-reflector/admin-state) that controls whether the device is allowed to respond to incoming TWAMP-Test sessions.

A device operating in the Session-Reflector role cannot configure attributes on a per-session basis, as it has no foreknowledge of what incoming sessions it will receive. As such, any parameter that the Session-Reflector might want to apply to an incoming TWAMP-Test session must be configured at the overall Session-Reflector level and are applied to all incoming sessions.

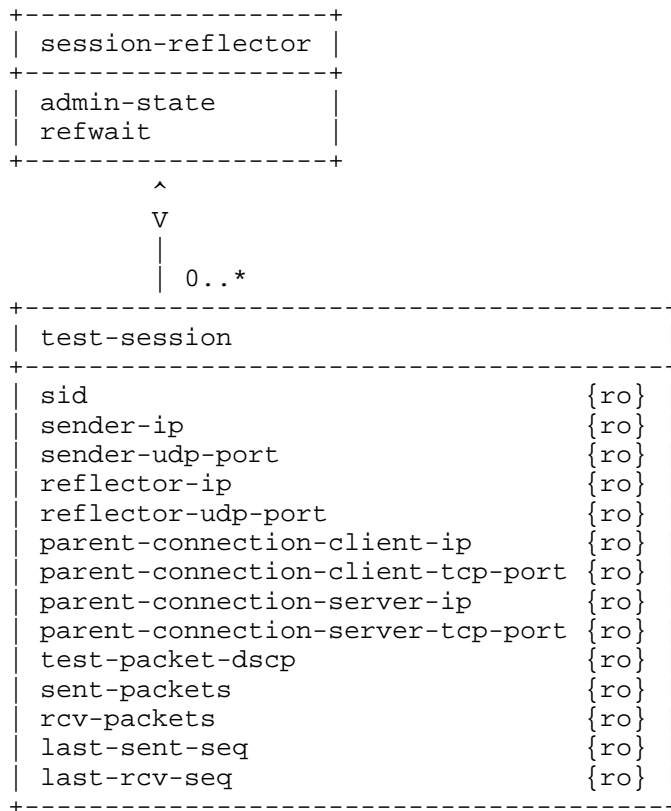


Figure 6: TWAMP Session-Reflector UML class diagram

Each incoming TWAMP-Test session that is active on the Session-Reflector SHALL be represented by an instance of a test-session object. All items in the test-session object are read-only.

Instances of test-session are indexed by a session identifier (sid). This value is auto-allocated by the TWAMP Server as test session requests are received, and communicated back to the Control-Client in the SID field of the Accept-Session message; see Section 4.3 of TWAMP Reflect Octets and Symmetrical Size Features [RFC6038].

When attempting to retrieve operational data for active test sessions from a Session-Reflector device, the user will not know what sessions are currently active on that device, or what SIDs have been auto-allocated for these test sessions. If the user has network access to the Control-Client device, then it is possible to read the data for this session under `client/ctrl-connection/test-session-request/sid` and obtain the SID (see Figure 3). The user may then use this SID

value as an index to retrieve an individual session-reflector/test-session instance on the Session-Reflector device.

If the user has no network access to the Control-Client device, then the only option is to retrieve all test-session instances from the Session-Reflector device, and then pick out specific test-session instances of interest to the user. This could be problematic if a large number of test sessions are currently active on that device.

Each Session-Reflector TWAMP-Test session contains the following 4-tuple: {parent-connection-client-ip, parent-connection-client-tcp-port, parent-connection-server-ip, parent-connection-server-tcp-port}. This 4-tuple MUST correspond to the equivalent 4-tuple {client-ip, client-tcp-port, server-ip, server-tcp-port} in server/ctrl-connection. This 4-tuple allows the user to trace back from the TWAMP-Test session to the (parent) TWAMP-Control connection that negotiated this test session.

5. Data Model

This section formally specifies the TWAMP data model using YANG.

5.1. YANG Tree Diagram

This section presents a simplified graphical representation of the TWAMP data model using a YANG tree diagram. Readers should keep in mind that the limit of 72 characters per line forces us to introduce artificial line breaks in some tree diagram nodes. Tree diagrams used in this document follow the notation defined in YANG Tree Diagrams [RFC8340].

```

module: ietf-twamp
  +--rw twamp
    +--rw client {control-client}?
      |
      |   +--rw admin-state?                boolean
      |   +--rw mode-preference-chain* [priority]
      |     |   +--rw priority            uint16
      |     |   +--rw mode?              twamp-modes
      |   +--rw key-chain* [key-id]
      |     |   +--rw key-id              string
      |     |   +--rw secret-key?       binary
      |   +--rw ctrl-connection* [name]
      |     +--rw name                    string
      |     +--rw client-ip?              inet:ip-address
      |     +--rw server-ip               inet:ip-address
      |     +--rw server-tcp-port?       inet:port-number
      |     +--rw control-packet-dscp?   inet:dscp
      |     +--rw key-id?                 string
  
```

```

|--rw max-count-exponent?      uint8
|--ro client-tcp-port?         inet:port-number
|--ro server-start-time?      uint64
|--ro repeat-count?           uint64
|--ro state?
|   control-client-connection-state
|--ro selected-mode?          twamp-modes
|--ro token?                  binary
|--ro client-iv?              binary
|--rw test-session-request* [name]
|   |--rw name                 string
|   |--rw sender-ip?          inet:ip-address
|   |--rw sender-udp-port?    union
|   |--rw reflector-ip       inet:ip-address
|   |--rw reflector-udp-port? inet:port-number
|   |--rw timeout?           uint64
|   |--rw padding-length?    uint32
|   |--rw test-packet-dscp?  inet:dscp
|   |--rw start-time?        uint64
|   |--rw repeat?            uint32
|   |--rw repeat-interval?   uint32
|   |--rw pm-reg-list* [pm-index]
|   |   |--rw pm-index      uint16
|   |--ro state?            test-session-state
|   |--ro sid?              string
+--rw server {server}?
|   |--rw admin-state?       boolean
|   |--rw server-tcp-port?   inet:port-number
|   |--rw servwait?         uint32
|   |--rw control-packet-dscp? inet:dscp
|   |--rw count?            uint8
|   |--rw max-count-exponent? uint8
|   |--rw modes?            twamp-modes
|   |--rw key-chain* [key-id]
|   |   |--rw key-id         string
|   |   |--rw secret-key?   binary
+--ro ctrl-connection*
|   [client-ip client-tcp-port server-ip server-tcp-port]
|   |--ro client-ip          inet:ip-address
|   |--ro client-tcp-port    inet:port-number
|   |--ro server-ip          inet:ip-address
|   |--ro server-tcp-port    inet:port-number
|   |--ro state?            server-ctrl-connection-state
|   |--ro control-packet-dscp? inet:dscp
|   |--ro selected-mode?    twamp-modes
|   |--ro key-id?           string
|   |--ro count?            uint8
|   |--ro max-count-exponent? uint8

```

```

    |         +--ro salt?                binary
    |         +--ro server-iv?           binary
    |         +--ro challenge?           binary
+--rw session-sender {session-sender}?
    |         +--rw admin-state?        boolean
    |         +--rw test-session* [name]
    |         |         +--rw name                string
    |         |         +--ro ctrl-connection-name? string
    |         |         +--rw fill-mode?        padding-fill-mode
    |         |         +--rw number-of-packets  uint32
    |         |         +--rw (packet-distribution)?
    |         |         |         +---:(periodic)
    |         |         |         |         +--rw periodic-interval    decimal64
    |         |         |         |         +---:(poisson)
    |         |         |         |         +--rw lambda                decimal64
    |         |         |         |         +--rw max-interval?       decimal64
    |         |         +--ro state?          sender-session-state
    |         |         +--ro sent-packets?   uint32
    |         |         +--ro rcv-packets?   uint32
    |         |         +--ro last-sent-seq?  uint32
    |         |         +--ro last-rcv-seq?  uint32
+--rw session-reflector {session-reflector}?
    |         +--rw admin-state?        boolean
    |         +--rw refwait?            uint32
    |         +--ro test-session*
    |         |         [sender-ip sender-udp-port reflector-ip reflector-udp
-port]
    |         |         +--ro sid?                string
    |         |         +--ro sender-ip          inet:ip-address
    |         |         +--ro sender-udp-port
    |         |         |         dynamic-port-number
    |         |         +--ro reflector-ip        inet:ip-address
    |         |         +--ro reflector-udp-port  inet:port-numbe
r
    |         +--ro parent-connection-client-ip?  inet:ip-address
    |         +--ro parent-connection-client-tcp-port?  inet:port-numbe
r
    |         +--ro parent-connection-server-ip?  inet:ip-address
    |         +--ro parent-connection-server-tcp-port?  inet:port-numbe
r
    |         +--ro test-packet-dscp?            inet:dscp
    |         +--ro sent-packets?               uint32
    |         +--ro rcv-packets?               uint32
    |         +--ro last-sent-seq?             uint32
    |         +--ro last-rcv-seq?             uint32

```

Figure 7: YANG Tree Diagram.

5.2. YANG Module

This section presents the YANG module for the TWAMP data model defined in this document. The module imports definitions from Common YANG Data Types [RFC6991], and references NTPv4 Specification [RFC5905], Framework for IP Performance Metrics [RFC2330], Randomness Requirements for Security [RFC4086], OWAMP [RFC4656], TWAMP [RFC5357], More Features for TWAMP [RFC5618], Individual Session Control Feature [RFC5938], TWAMP Reflect Octets and Symmetrical Size Features [RFC6038], Advances Stream and Sampling Framework [RFC7312], IKEv2-Derived Shared Secret Key for OWAMP and TWAMP [RFC7717], and OWAMP and TWAMP Well-Known Port Assignments [I-D.ietf-ippm-port-twamp-test].

```
<CODE BEGINS> file "ietf-twamp@2018-07-02.yang"

module ietf-twamp {
  yang-version 1.1;
  namespace urn:ietf:params:xml:ns:yang:ietf-twamp;
  prefix ietf-twamp;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }

  organization
    "IETF IPPM (IP Performance Metrics) Working Group";

  contact
    "WG Web: http://tools.ietf.org/wg/ippm/
    WG List: ippm@ietf.org

    Editor: Ruth Civil
            gcivil@ciena.com
    Editor: Al Morton
            acmorton@att.com
    Editor: Reshad Rehman
            rrahman@cisco.com
    Editor: Mahesh Jethanandani
            mjethanandani@gmail.com
    Editor: Kostas Pentikousis
            k.pentikousis@travelping.com";

  description
    "This YANG module specifies a vendor-independent data
```

model for the Two-Way Active Measurement Protocol (TWAMP).

The data model covers four TWAMP logical entities, namely, Control-Client, Server, Session-Sender, and Session-Reflector, as illustrated in the annotated TWAMP logical model (Fig. 1 of RFC XXXX).

This YANG module uses features to indicate which of the four logical entities are supported by a TWAMP implementation.

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2018-07-02 {
  description
    "Initial Revision.

    Covers RFC 5357, RFC 5618, RFC 5938, RFC 6038, RFC 7717, and
    draft-ietf-ippm-metric-registry";

  reference
    "RFC XXXX: TWAMP YANG Data Model.";
}
```

```
/*
 * Typedefs
 */
```

```
typedef twamp-modes {
  type bits {
    bit unauthenticated {
      position 0;
      description
        "Unauthenticated mode, in which no encryption or
        authentication is applied in TWAMP-Control and
        TWAMP-Test. KeyID, Token, and Client-IV are not used in
        the Set-Up-Response message. See Section 3.1 of
        RFC 4656.";
```

```
reference
  "RFC 4656: A One-way Active Measurement Protocol
  (OWAMP)";
}
bit authenticated {
  position 1;
  description
    "Authenticated mode, in which the Control-Client and
    Server possess a shared secret thus prohibiting
    'theft of service'. As per Section 6 of RFC 4656,
    in 'authenticated mode, the timestamp is in the clear
    and is not protected cryptographically in any way,
    while the rest of the message has the same protection
    as in encrypted mode. This mode allows one to trade off
    cryptographic protection against accuracy of
    timestamps.'"
  reference
    "RFC 4656: A One-way Active Measurement Protocol
    (OWAMP)";
}
bit encrypted {
  position 2;
  description
    "Encrypted mode 'makes it impossible to alter
    timestamps undetectably' [Section 6 of RFC 4656].
    See also Section 4 of RFC 7717."
  reference
    "RFC 4656: A One-way Active Measurement Protocol
    (OWAMP)";
}
bit unauth-test-encrypt-control {
  position 3;
  description
    "When using the Mixed Security Mode, the TWAMP-Test
    protocol follows the Unauthenticated mode and the
    TWAMP-Control protocol the Encrypted mode."
  reference
    "RFC 5618: Mixed Security Mode for the Two-Way Active
    Measurement Protocol (TWAMP)";
}
bit individual-session-control {
  position 4;
  description
    "This mode enables individual test sessions using
    Session Identifiers."
  reference
    "RFC 5938: Individual Session Control Feature
    for the Two-Way Active Measurement Protocol (TWAMP)";
```



```
    }
    bit reflect-octets {
      position 5;
      description
        "This mode indicates the reflect octets capability.";
      reference
        "RFC 6038: Two-Way Active Measurement Protocol (TWAMP)
        Reflect Octets and Symmetrical Size Features";
    }
    bit symmetrical-size {
      position 6;
      description
        "This mode indicates support for the symmetrical size
        sender test packet format.";
      reference
        "RFC 6038: Two-Way Active Measurement Protocol (TWAMP)
        Reflect Octets and Symmetrical Size Features";
    }
    bit IKEv2Derived {
      position 7;
      description
        "In this mode the the shared key is derived
        from an IKEv2 security association (SA).";
      reference
        "RFC 7717: IKEv2-Derived Shared Secret Key for
        the One-Way Active Measurement Protocol (OWAMP)
        and Two-Way Active Measurement Protocol (TWAMP)";
    }
  }
  description
    "Specifies the configurable TWAMP-Modes supported during a
    TWAMP-Control Connection setup between a Control-Client
    and a Server. Section 7 of RFC 7717 summarizes the
    TWAMP-Modes registry and points to their formal
    specification.";
}

typedef control-client-connection-state {
  type enumeration {
    enum active {
      description
        "Indicates an active TWAMP-Control connection to
        Server.";
    }
    enum idle {
      description
        "Indicates an idle TWAMP-Control connection to Server.";
    }
  }
}
```

```
    }
    description
      "Indicates the Control-Client TWAMP-Control connection
       state.";
  }

typedef test-session-state {
  type enumeration {
    enum accepted {
      value 0;
      description
        "Indicates an accepted TWAMP-Test session request.";
    }
    enum failed {
      value 1;
      description
        "Indicates a TWAMP-Test session failure due to
         some unspecified reason (catch-all).";
    }
    enum internal-error {
      value 2;
      description
        "Indicates a TWAMP-Test session failure due to
         an internal error.";
    }
    enum not-supported {
      value 3;
      description
        "Indicates a TWAMP-Test session failure because
         some aspect of the TWAMP-Test session request
         is not supported.";
    }
    enum permanent-resource-limit {
      value 4;
      description
        "Indicates a TWAMP-Test session failure due to
         permanent resource limitations.";
    }
    enum temp-resource-limit {
      value 5;
      description
        "Indicates a TWAMP-Test session failure due to
         temporary resource limitations.";
    }
  }
  description
    "Indicates the Control-Client TWAMP-Test session state.";
}
```

```
typedef server-ctrl-connection-state {
  type enumeration {
    enum active {
      description
        "Indicates an active TWAMP-Control connection
        to the Control-Client.";
    }
    enum servwait {
      description
        "Indicates that the TWAMP-Control connection to the
        Control-Client is in SERVWAIT as per the definition of
        Section 3.1 of RFC 5357.";
    }
  }
  description
    "Indicates the Server TWAMP-Control connection state.";
}

typedef sender-session-state {
  type enumeration {
    enum active {
      description
        "Indicates that the TWAMP-Test session is active.";
    }
    enum failure {
      description
        "Indicates that the TWAMP-Test session has failed.";
    }
  }
  description
    "Indicates the Session-Sender TWAMP-Test session state.";
}

typedef padding-fill-mode {
  type enumeration {
    enum zero {
      description
        "TWAMP-Test packets are padded with all zeros.";
    }
    enum random {
      description
        "TWAMP-Test packets are padded with pseudo-random
        numbers.";
    }
  }
  description
    "Indicates what type of packet padding is used in the
    TWAMP-Test packets.";
}
```

```
    }

    typedef dynamic-port-number {
      type inet:port-number {
        range 49152..65535;
      }
      description "Dynamic range for port numbers.";
    }

    /*
     * Features
     */

    feature control-client {
      description
        "Indicates that the device supports configuration of the
        TWAMP Control-Client logical entity.";
    }

    feature server {
      description
        "Indicates that the device supports configuration of the
        TWAMP Server logical entity.";
    }

    feature session-sender {
      description
        "Indicates that the device supports configuration of the
        TWAMP Session-Sender logical entity.";
    }

    feature session-reflector {
      description
        "Indicates that the device supports configuration of the
        TWAMP Session-Reflector logical entity.";
    }

    /*
     * Reusable node groups
     */

    grouping key-management {
      list key-chain {
        key key-id;
        leaf key-id {
          type string {
            length 1..80;
          }
        }
      }
    }
  }
}
```

```
    }
    description
      "KeyID used for a TWAMP-Control connection. As per
      Section 3.1 of RFC 4656, KeyID is 'a UTF-8 string, up to
      80 octets in length' and is used to select which 'shared
      shared secret the [Control-Client] wishes to use to
      authenticate or encrypt'.";
    }
    leaf secret-key {
      type binary;
      description
        "The secret key corresponding to the KeyID for this
        TWAMP-Control connection.";
    }
    description
      "Relates KeyIDs with their respective secret keys
      in a TWAMP-Control connection.";
  }
  description
    "Used by the Control-Client and Server for TWAMP-Control
    key management.";
}

grouping maintenance-statistics {
  leaf sent-packets {
    type uint32;
    config false;
    description
      "Indicates the number of packets sent.";
  }

  leaf rcv-packets {
    type uint32;
    config false;
    description
      "Indicates the number of packets received.";
  }

  leaf last-sent-seq {
    type uint32;
    config false;
    description
      "Indicates the last sent sequence number.";
  }

  leaf last-rcv-seq {
    type uint32;
    config false;
  }
}
```

```
        description
          "Indicates the last received sequence number.";
      }
      description
        "Used for TWAMP-Test maintenance statistics.";
    }

    grouping count {
      leaf count {
        type uint8 {
          range "10..31";
        }
        default 15;
        description
          "Parameter communicated to the Control-Client as part of
           the Server Greeting message and used for deriving a key
           from a shared secret as per Section 3.1 of RFC 4656:
           MUST be a power of 2 and at least 1024. It is configured
           by providing said power. For example, configuring 20 here
           means count  $2^{20} = 1048576$ . The default is 15,
           meaning  $2^{15} = 32768$ .";
        }
      description
        "Reusable data structure for count, which is used both in the
         Server and the Control-Client.";
    }

    grouping max-count-exponent {
      leaf max-count-exponent {
        type uint8 {
          range 10..31;
        }
        default 20;
        description
          "This parameter limits the maximum Count value, which MUST
           be a power of 2 and at least 1024 as per RFC 5357. It is
           configured by providing said power. For example,
           configuring 10 here means max count  $2^{10} = 1024$ .
           The default is 20, meaning  $2^{20} = 1048576$ .

           A TWAMP Server uses this configured value in the
           Server-Greeting message sent to the Control-Client.

           A TWAMP Control-Client uses this configured value to
           prevent denial-of-service (DOS) attacks by closing the
           control connection to the Server if it 'receives a
           Server-Greeting message with Count greater than its
           maximum configured value', as per Section 6 of RFC 5357.
```

Further, note that according to Section 6 of RFC 5357:

'If an attacking system sets the maximum value in Count (2^{32}), then the system under attack would stall for a significant period of time while it attempts to generate keys.

TWAMP-compliant systems SHOULD have a configuration control to limit the maximum count value. The default max-count-exponent value SHOULD be 15 which corresponds to a maximum value of 2^{15} or 32768.'

RFC 5357 does not qualify 'significant period' in terms of time, but it is clear that this depends on the processing capacity available and operators need to pay attention to this security consideration.";

```
    }
  description
    "Reusable data structure for max-count which is used both at
    the Control-Client and the Server containers.";
}

/*
 * Configuration data nodes
 */

container twamp {
  description
    "TWAMP logical entity configuration grouping of four models
    which correspond to the four TWAMP logical entities
    Control-Client, Server, Session-Sender, and Session-Reflector
    as illustrated in Fig. 1 of RFC XXXX.";

  container client {
    if-feature control-client;
    description
      "Configuration of the TWAMP Control-Client logical
      entity.";

    leaf admin-state {
      type boolean;
      default true;
      description
        "Indicates whether the device is allowed to operate as a
        TWAMP Control-Client.";
    }
  }
}
```

```
list mode-preference-chain {
  key priority;
  unique mode;
  leaf priority {
    type uint16;
    description
      "Indicates the Control-Client Mode preference priority
       expressed as a 16-bit unsigned integer. Values for the
       priority start with zero, the highest priority, and
       decreasing priority value is indicated by every increase
       in value by one.";
  }
  leaf mode {
    type twamp-modes;
    description
      "The supported TWAMP Mode matching the corresponding
       priority.";
  }
  description
    "Indicates the Control-Client preferred order of use of
     the supported TWAMP Modes.

     Depending on the Modes available in the TWAMP Server
     Greeting message (see Fig. 2 of RFC 7717), the
     Control-Client MUST choose the highest priority
     Mode from the configured mode-preference-chain list.";
}

uses key-management;

list ctrl-connection {
  key name;
  description
    "List of TWAMP Control-Client control connections.
     Each item in the list describes a control connection
     that will be initiated by this Control-Client";

  leaf name {
    type string;
    description
      "A unique name used as a key to identify this
       individual TWAMP-Control connection on the
       Control-Client device.";
  }
  leaf client-ip {
    type inet:ip-address;
    description
```



```
        "The IP address of the local Control-Client device,
        to be placed in the source IP address field of the
        IP header in TWAMP-Control (TCP) packets belonging
        to this control connection. If not configured, the
        device SHALL choose its own source IP address.";
    }
    leaf server-ip {
        type inet:ip-address;
        mandatory true;
        description
            "The IP address of the remote Server device, which the
            TWAMP-Control connection will be initiated to.";
    }

    leaf server-tcp-port {
        type inet:port-number;
        default 862;
        description
            "This parameter defines the TCP port number that is
            to be used by this outgoing TWAMP-Control connection.
            Typically, this is the well-known TWAMP-Control
            port number (862) as per RFC 5357. However, there are
            known realizations of TWAMP in the field that were
            implemented before this well-known port number was
            allocated. These early implementations allowed the
            port number to be configured. This parameter is
            therefore provided for backward compatibility
            reasons.";
    }

    leaf control-packet-dscp {
        type inet:dscp;
        default 0;
        description
            "The DSCP value to be placed in the IP header of
            TWAMP-Control (TCP) packets generated by this
            Control-Client.";
    }

    leaf key-id {
        type string {
            length 1..80;
        }
        description
            "Indicates the KeyID value selected for this
            TWAMP-Control connection.";
    }
}
```

```
uses max-count-exponent;

leaf client-tcp-port {
  type inet:port-number;
  config false;
  description
    "Indicates the source TCP port number used in the
    TWAMP-Control packets belonging to this control
    connection.";
}

leaf server-start-time {
  type uint64;
  config false;
  description
    "Indicates the Start-Time advertised by the Server in
    the Server-Start message (RFC 4656, Section 3.1),
    representing the time when the current
    instantiation of the Server started operating.
    The timestamp format follows RFC 5905
    according to Section 4.1.2 of RFC 4656.";
  reference
    "RFC 4656: OWAMP, Section 3.1 and 4.1.2,
    RFC 5905: NTPv4 Specification.";
}

leaf repeat-count {
  type uint64;
  config false;
  description
    "Indicates how many times the test session has been
    repeated. When a test is running, this value will be
    greater than 0. If the repeat parameter is non-zero,
    this value is smaller than or equal to the repeat
    parameter.";
}

leaf state {
  type control-client-connection-state;
  config false;
  description
    "Indicates the current state of the TWAMP-Control
    connection state.";
}

leaf selected-mode {
  type twamp-modes;
  config false;
  description
```

```
    "The TWAMP Mode that the Control-Client has chosen for
    this control connection as set in the Mode field of
    the Set-Up-Response message";
  reference
    "RFC 4656, Section 3.1.";
}

leaf token {
  type binary {
    length 64;
  }
  config false;
  description
    "This parameter holds the 64 octets containing the
    concatenation of a 16-octet Challenge, a 16-octet AES
    Session-key used for encryption, and a 32-octet
    HMAC-SHA1 Session-key used for authentication; see
    also the last paragraph of Section 6 in RFC 4656.

    If the Mode defined in RFC 7717 is selected
    (selected-mode), Token is limited to 16 octets.";
  reference
    "RFC 4086: Randomness Requirements for Security

    RFC 7717: IKEv2-Derived Shared Secret Key for the
    One-Way Active Measurement Protocol (OWAMP) and
    Two-Way Active Measurement Protocol (TWAMP)";
}

leaf client-iv {
  type binary {
    length 16;
  }
  config false;
  description
    "Indicates the Control-Client Initialization Vector
    (Client-IV), that is generated randomly by the
    Control-Client. As per RFC 4656:

    Client-IV merely needs to be unique (i.e., it MUST
    never be repeated for different sessions using the
    same secret key; a simple way to achieve that without
    the use of cumbersome state is to generate the
    Client-IV values using a cryptographically secure
    pseudo-random number source.

    If the Mode defined in RFC 7717 is selected
    (selected-mode), Client-IV is limited to 12 octets.";
```

```
reference
  "RFC 4656: A One-way Active Measurement Protocol
  (OWAMP).

  RFC 7717: IKEv2-Derived Shared Secret Key for the
  One-Way Active Measurement Protocol (OWAMP) and
  Two-Way Active Measurement Protocol (TWAMP)";
}

list test-session-request {
  key name;
  description
    "Information associated with the Control-Client
    for this test session";

  leaf name {
    type string;
    description
      "A unique name to be used for identification of
      this TWAMP-Test session on the Control-Client.";
  }

  leaf sender-ip {
    type inet:ip-address;
    description
      "The IP address of the Session-Sender device,
      which is to be placed in the source IP address
      field of the IP header in TWAMP-Test (UDP) packets
      belonging to this test session. This value will be
      used to populate the sender address field of the
      Request-TW-Session message.

      If not configured, the device SHALL choose its own
      source IP address.";
  }

  leaf sender-udp-port {
    type union {
      type dynamic-port-number;
      type enumeration {
        enum autoallocate {
          description
            "Indicates that the Contol-Client will
            auto-allocate the TWAMP-Test (UDP) port number
            from the dynamic port range.";
        }
      }
    }
  }
}
```

```
default autoallocate;
description
  "The UDP port number that is to be used by
  the Session-Sender for this TWAMP-Test session.
  The number is restricted to the dynamic port range.

  By default the Control-Client SHALL auto-allocate a
  UDP port number for this TWAMP-Test session.

  The configured (or auto-allocated) value is
  advertised in the Sender Port field of the
  Request-TW-session message (see Section 3.5 of
  RFC 5357). Note that in the scenario where a device
  auto-allocates a UDP port number for a session, and
  the repeat parameter for that session indicates that
  it should be repeated, the device is free to
  auto-allocate a different UDP port number when it
  negotiates the next (repeated) iteration of this
  session."
}

leaf reflector-ip {
  type inet:ip-address;
  mandatory true;
  description
    "The IP address belonging to the remote
    Session-Reflector device to which the TWAMP-Test
    session will be initiated. This value will be
    used to populate the receiver address field of
    the Request-TW-Session message."
}

leaf reflector-udp-port {
  type inet:port-number {
    range "862 | 49152..65535";
  }
  description
    "This parameter defines the UDP port number that
    will be used by the Session-Reflector for
    this TWAMP-Test session. The default number is
    within the dynamic port range and is to be placed
    in the Receiver Port field of the Request-TW-Session
    message. The well-known port (862) MAY be
    used."
  reference
    "draft-ietf-ippm-port-twamp-test: OWAMP and TWAMP
    Well-Known Port Assignments."
}
```

```
leaf timeout {
  type uint64;
  units seconds;
  default 2;
  description
    "The length of time (in seconds) that the
    Session-Reflector should continue to respond to
    packets belonging to this TWAMP-Test session after
    a Stop-Sessions TWAMP-Control message has been
    received.

    This value will be placed in the Timeout field of
    the Request-TW-Session message.";
  reference
    "RFC 5357: TWAMP, Section 3.5.";
}

leaf padding-length {
  type uint32 {
    range 64..4096;
  }
  description
    "The number of padding bytes to be added to the
    TWAMP-Test (UDP) packets generated by the
    Session-Sender.

    This value will be placed in the Padding Length
    field of the Request-TW-Session message.";
  reference
    "RFC 4656, Section 3.5.";
}

leaf test-packet-dscp {
  type inet:dscp;
  default 0;
  description
    "The DSCP value to be placed in the IP header
    of TWAMP-Test packets generated by the
    Session-Sender, and in the UDP header of the
    TWAMP-Test response packets generated by the
    Session-Reflector for this test session.

    This value will be placed in the Type-P Descriptor
    field of the Request-TW-Session message";
  reference
    "RFC 5357.";
}
```

```
leaf start-time {
  type uint64;
  default 0;
  description
    "Time when the session is to be started
    (but not before the TWAMP Start-Sessions command
    is issued; see Section 3.4 of RFC 5357).

    The start-time value is placed in the Start Time
    field of the Request-TW-Session message.

    The timestamp format follows RFC 5905 as per
    Section 3.5 of RFC 4656.

    The default value of 0 indicates that the session
    will be started as soon as the Start-Sessions
    message is received.";
}

leaf repeat {
  type uint32 {
    range 0..4294967295;
  }
  default 0;
  description
    "This value determines if the TWAMP-Test session must
    be repeated. When a test session has completed, the
    repeat parameter is checked.

    The default value of 0 indicates that the session
    MUST NOT be repeated.

    If the repeat value is 1 through 4,294,967,294
    then the test session SHALL be repeated using the
    information in repeat-interval parameter, and the
    parent TWAMP-Control connection for this test
    session is restarted to negotiate a new instance
    of this TWAMP-Test session.

    A value of 4,294,967,295 indicates that the test
    session SHALL be repeated *forever* using the
    information in repeat-interval parameter, and SHALL
    NOT decrement the value.";
}

leaf repeat-interval {
  when "../repeat!='0'" {
    description
```

```
    "This parameter determines the timing of repeated
    TWAMP-Test sessions when repeat is more than 0.

    When the value of repeat-interval is 0, the
    negotiation of a new test session SHALL begin
    immediately after the previous test session
    completes. Otherwise, the Control-Client will
    wait for the number of seconds specified in the
    repeat-interval parameter before negotiating the
    new instance of this TWAMP-Test session.";
  }
  type uint32;
  units seconds;
  default 0;
  description
    "Repeat interval (in seconds).";
}

list pm-reg-list {
  key pm-index;
  leaf pm-index {
    type uint16;
    description
      "Numerical index value of a Registered Metric
      in the Performance Metric Registry
      (see ietf-ippm-metric-registry). Output statistics
      are specified in the corresponding Registry
      entry.";
  }
  description
    "A list of one or more Performance Metric Registry
    Index values, which communicate packet stream
    characteristics along with one or more metrics
    to be measured.

    All members of the pm-reg-list MUST have the same
    stream characteristics, such that they combine
    to specify all metrics that shall be measured on
    a single stream.";
  reference
    "ietf-ippm-metric-registry: Registry for
    Performance Metrics";
}

leaf state {
  type test-session-state;
  config false;
  description
```



```
        "Indicates the TWAMP-Test session state, accepted or
        indication of an error.";
    reference
        "Section 3.5 of RFC 5357.";
    }
    leaf sid {
        type string;
        config false;
        description
            "The SID allocated by the Server for this TWAMP-Test
            session, and communicated back to the Control-Client
            in the SID field of the Accept-Session message";
        reference
            "Section 4.3 of RFC 6038.";
    }
    }
}

container server {
    if-feature server;
    description
        "Configuration of the TWAMP Server logical entity.";

    leaf admin-state {
        type boolean;
        default true;
        description
            "Indicates whether the device is allowed to operate
            as a TWAMP Server.";
    }

    leaf server-tcp-port {
        type inet:port-number;
        default 862;
        description
            "This parameter defines the well known TCP port number
            that is used by TWAMP-Control. The Server will listen
            on this port number for incoming TWAMP-Control
            connections. Although this is defined as a fixed value
            (862) in RFC 5357, there are several realizations of
            TWAMP in the field that were implemented before this
            well-known port number was allocated. These early
            implementations allowed the port number to be
            configured. This parameter is therefore provided for
            backward compatibility reasons.";
    }
}
```

```
leaf servwait {
  type uint32 {
    range 1..604800;
  }
  units seconds;
  default 900;
  description
    "TWAMP-Control (TCP) session timeout, in seconds.
    According to Section 3.1 of RFC 5357,

    Server MAY discontinue any established control
    connection when no packet associated with that
    connection has been received within SERVWAIT seconds.";
}

leaf control-packet-dscp {
  type inet:dscp;
  description
    "The DSCP value to be placed in the IP header of
    TWAMP-Control (TCP) packets generated by the Server.

    Section 3.1 of RFC 5357 specifies that the server
    SHOULD use the DSCP value from the Control-Clients
    TCP SYN. However, for practical purposes TWAMP will
    typically be implemented using a general purpose TCP
    stack provided by the underlying operating system,
    and such a stack may not provide this information to the
    user. Consequently, it is not always possible to
    implement the behavior described in RFC 5357 in an
    OS-portable version of TWAMP.

    The default behavior if this item is not set is to use
    the DSCP value from the Control-Clients TCP SYN.";
  reference
    "Section 3.1 of RFC 5357.";
}

uses count;

uses max-count-exponent;

leaf modes {
  type twamp-modes;
  description
    "The bit mask of TWAMP Modes this Server instance
    is willing to support; see IANA TWAMP Modes Registry.";
}
```

```
uses key-management;

list ctrl-connection {
  key "client-ip client-tcp-port server-ip server-tcp-port";
  config false;
  description
    "List of all incoming TWAMP-Control (TCP) connections.";

  leaf client-ip {
    type inet:ip-address;
    description
      "The IP address on the remote Control-Client device,
      which is the source IP address used in the
      TWAMP-Control (TCP) packets belonging to this control
      connection.";
  }

  leaf client-tcp-port {
    type inet:port-number;
    description
      "The source TCP port number used in the TWAMP-Control
      (TCP) packets belonging to this control connection.";
  }

  leaf server-ip {
    type inet:ip-address;
    description
      "The IP address of the local Server device, which is
      the destination IP address used in the
      TWAMP-Control (TCP) packets belonging to this control
      connection.";
  }

  leaf server-tcp-port {
    type inet:port-number;
    description
      "The destination TCP port number used in the
      TWAMP-Control (TCP) packets belonging to this
      control connection. This will usually be the
      same value as the server-tcp-port configured
      under twamp/server. However, in the event that
      the user re-configured server/server-tcp-port
      after this control connection was initiated, this
      value will indicate the server-tcp-port that is
      actually in use for this control connection.";
  }

  leaf state {
```

```
    type server-ctrl-connection-state;
    description
      "Indicates the Server TWAMP-Control connection state.";
  }

  leaf control-packet-dscp {
    type inet:dscp;
    description
      "The DSCP value used in the IP header of the
      TWAMP-Control (TCP) packets sent by the Server
      for this control connection. This will usually
      be the same value as is configured in the
      control-packet-dscp parameter under the twamp/server
      container. However, in the event that the user
      re-configures server/dscp after this control
      connection is already in progress, this read-only
      value will show the actual dscp value in use by this
      TWAMP-Control connection.";
  }

  leaf selected-mode {
    type twamp-modes;
    description
      "The Mode that was chosen for this TWAMP-Control
      connection as set in the Mode field of the
      Set-Up-Response message.";
  }

  leaf key-id {
    type string {
      length 1..80;
    }
    description
      "The KeyID value that is in use by this TWAMP-Control
      connection as selected by Control-Client.";
  }

  uses count {
    description
      "The count value that is in use by this TWAMP-Control
      connection. This will usually be the same value
      as is configured under twamp/server. However, in the
      event that the user re-configured server/count
      after this control connection is already in progress,
      this read-only value will show the actual count that
      is in use for this TWAMP-Control connection.";
  }
}
```

```
    uses max-count-exponent {
      description
        "This read-only value indicates the actual max-count in
        use for this control connection. Usually this would be
        the same value as configured under twamp/server.";
    }

    leaf salt {
      type binary {
        length 16;
      }
      description
        "A parameter used in deriving a key from a
        shared secret as described in Section 3.1 of RFC 4656.
        It is communicated to the Control-Client as part of
        the Server Greeting message.";
    }

    leaf server-iv {
      type binary {
        length 16;
      }
      description
        "The Server Initialization Vector
        (IV) generated randomly by the Server.";
    }

    leaf challenge {
      type binary {
        length 16;
      }
      description
        "A random sequence of octets generated by the Server.
        As described in client/token, Challenge is used
        by the Control-Client to prove possession of a
        shared secret.";
    }
  }
}

container session-sender {
  if-feature session-sender;
  description
    "Configuration of the TWAMP Session-Sender logical entity";
  leaf admin-state {
    type boolean;
    default true;
    description
```

```
        "Indicates whether the device is allowed to operate
        as a TWAMP Session-Sender.";
    }

list test-session{
    key name;
    description
        "List of TWAMP Session-Sender test sessions.";

    leaf name {
        type string;
        description
            "A unique name for this TWAMP-Test session to be used
            for identifying this test session by the
            Session-Sender logical entity.";
    }

    leaf ctrl-connection-name {
        type string;
        config false;
        description
            "The name of the parent TWAMP-Control connection that
            is responsible for negotiating this TWAMP-Test
            session.";
    }

    leaf fill-mode {
        type padding-fill-mode;
        default zero;
        description
            "Indicates whether the padding added to the
            TWAMP-Test (UDP) packets will contain pseudo-random
            numbers, or whether it should consist of all zeroes,
            as per Section 4.2.1 of RFC 5357.";
    }

    leaf number-of-packets {
        type uint32;
        mandatory true;
        description
            "The overall number of TWAMP-Test (UDP) packets to be
            transmitted by the Session-Sender for this test
            session.";
    }

    choice packet-distribution {
        description
            "Indicates the distribution to be used for transmitting
```

```
    the TWAMP-Test (UDP) packets.";
  case periodic {
    leaf periodic-interval {
      type decimal64 {
        fraction-digits 5;
      }
      units seconds;
      mandatory true;
      description
        "Indicates the time to wait (in seconds) between
         the first bits of TWAMP-Test (UDP) packet
         transmissions for this test session.";
      reference
        "RFC 3432: Network performance measurement
         with periodic streams";
    }
  }
  case poisson {
    leaf lambda {
      type decimal64 {
        fraction-digits 5;
      }
      units seconds;
      mandatory true;
      description
        "Indicates the average time interval (in seconds)
         between packets in the Poisson distribution.
         The packet is calculated using the reciprocal of
         lambda and the TWAMP-Test packet size (which
         depends on the selected Mode and the packet
         padding).";
      reference
        "RFC 2330: Framework for IP Performance Metrics";
    }
    leaf max-interval {
      type decimal64 {
        fraction-digits 5;
      }
      units seconds;
      description
        "Indicates the maximum time (in seconds)
         between packet transmissions.";
      reference
        "RFC 7312: Advanced Stream and Sampling Framework
         for IP Performance Metrics (IPPM)";
    }
  }
}
```

```
leaf state {
  type sender-session-state;
  config false;
  description
    "Indicates the Session-Sender test session state.";
}

uses maintenance-statistics;
}
}

container session-reflector {
  if-feature session-reflector;
  description
    "Configuration of the TWAMP Session-Reflector logical
    entity";

  leaf admin-state {
    type boolean;
    default true;
    description
      "Indicates whether the device is allowed to operate
      as a TWAMP Session-Reflector.";
  }

  leaf refwait {
    type uint32 {
      range 1..604800;
    }
    units seconds;
    default 900;
    description
      "The Session-Reflector MAY discontinue any session that
      has been started when no packet associated with that
      session has been received for REFWAIT seconds. As per
      Section 3.1 of RFC 5357, this timeout allows a
      Session-Reflector to free up resources in case of
      failure.";
  }

  list test-session {
    key
      "sender-ip sender-udp-port
      reflector-ip reflector-udp-port";
    config false;
    description
      "TWAMP Session-Reflector test sessions.";
  }
}
```



```
leaf sid {
  type string;
  description
    "An auto-allocated identifier for this TWAMP-Test
    session that is unique within the context of this
    Server/Session-Reflector device only. This value
    is communicated to the Control-Client that
    requested the test session in the SID field of the
    Accept-Session message.";
}

leaf sender-ip {
  type inet:ip-address;
  description
    "The IP address on the remote device, which is the
    source IP address used in the TWAMP-Test (UDP) packets
    belonging to this test session.";
}

leaf sender-udp-port {
  type dynamic-port-number;
  description
    "The source UDP port used in the TWAMP-Test packets
    belonging to this test session.";
}

leaf reflector-ip {
  type inet:ip-address;
  description
    "The IP address of the local Session-Reflector
    device, which is the destination IP address used
    in the TWAMP-Test (UDP) packets belonging to this test
    session.";
}

leaf reflector-udp-port {
  type inet:port-number {
    range "862 | 49152..65535";
  }
  description
    "The destination UDP port number used in the
    TWAMP-Test (UDP) test packets belonging to this
    test session.";
}

leaf parent-connection-client-ip {
  type inet:ip-address;
  description
```

```
        "The IP address on the Control-Client device, which
        is the source IP address used in the TWAMP-Control
        (TCP) packets belonging to the parent control
        connection that negotiated this test session.";
    }

    leaf parent-connection-client-tcp-port {
        type inet:port-number;
        description
            "The source TCP port number used in the TWAMP-Control
            (TCP) packets belonging to the parent control
            connection that negotiated this test session.";
    }

    leaf parent-connection-server-ip {
        type inet:ip-address;
        description
            "The IP address of the Server device, which is the
            destination IP address used in the TWAMP-Control
            (TCP) packets belonging to the parent control
            connection that negotiated this test session.";
    }

    leaf parent-connection-server-tcp-port {
        type inet:port-number;
        description
            "The destination TCP port number used in the
            TWAMP-Control (TCP) packets belonging to the parent
            control connection that negotiated this test
            session.";
    }

    leaf test-packet-dscp {
        type inet:dscp;
        description
            "The DSCP value present in the IP header of
            TWAMP-Test (UDP) packets belonging to this session.";
    }

    uses maintenance-statistics;
}
}
}
```

<CODE ENDS>

6. Data Model Examples

This section presents a simple but complete example of configuring all four entities in Figure 1, based on the YANG module specified in Section 5. The example is illustrative in nature, but aims to be self-contained, i.e. were it to be executed in a real TWAMP implementation it would lead to a correctly configured test session. For completeness, examples are provided for both IPv4 and IPv6.

A more elaborated example, which also includes authentication parameters, is provided in Appendix A.

6.1. Control-Client

Figure 8 shows a configuration example for a Control-Client with client/admin-state enabled. In a real implementation following Figure 2 this would permit the initiation of TWAMP-Control connections and TWAMP-Test sessions.

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
    </client>
  </twamp>
</config>
```

Figure 8: XML instance enabling Control-Client operation.

The following example shows a Control-Client with two instances of client/ctrl-connection, one called "RouterA" and another called "RouterB". Each TWAMP-Control connection is to a different Server. The control connection named "RouterA" has two test session requests. The TWAMP-Control connection named "RouterB" has no TWAMP-Test session requests.

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <name>RouterA</name>
        <client-ip>203.0.113.1</client-ip>
        <server-ip>203.0.113.2</server-ip>
        <test-session-request>
```

```

        <name>Test1</name>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <start-time>0</start-time>
    </test-session-request>
    <test-session-request>
        <name>Test2</name>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>203.0.113.2</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <start-time>0</start-time>
    </test-session-request>
</ctrl-connection>
<ctrl-connection>
    <name>RouterB</name>
    <client-ip>203.0.113.1</client-ip>
    <server-ip>203.0.113.3</server-ip>
</ctrl-connection>
</client>
</twamp>
</config>

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <name>RouterA</name>
        <client-ip>2001:DB8:203:0:113::1</client-ip>
        <server-ip>2001:DB8:203:0:113::2</server-ip>
        <test-session-request>
          <name>Test1</name>
          <sender-ip>2001:DB8:203:1:113::3</sender-ip>
          <sender-udp-port>54000</sender-udp-port>
          <reflector-ip>2001:DB8:203:1:113::4</reflector-ip>
          <reflector-udp-port>55000</reflector-udp-port>
          <start-time>0</start-time>
        </test-session-request>
        <test-session-request>
          <name>Test2</name>
          <sender-ip>2001:DB8:203:0:113::1</sender-ip>
          <sender-udp-port>54001</sender-udp-port>
          <reflector-ip>2001:DB8:203:0:113::2</reflector-ip>
          <reflector-udp-port>55001</reflector-udp-port>
        </test-session-request>
      </ctrl-connection>
    </client>
  </twamp>
</config>

```

```
        <start-time>0</start-time>
      </test-session-request>
    </ctrl-connection>
    <ctrl-connection>
      <name>RouterB</name>
      <client-ip>2001:DB8:203:0:113::1</client-ip>
      <server-ip>2001:DB8:203:0:113::3</server-ip>
    </ctrl-connection>
  </client>
</twamp>
</config>
```

6.2. Server

Figure 9 shows a configuration example for a Server with `server/admin-state` enabled, which permits a device following Figure 2 to respond to TWAMP-Control connections and TWAMP-Test sessions.

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
    </server>
  </twamp>
</config>
```

Figure 9: XML instance enabling Server operation.

The following example presents a Server with the TWAMP-Control connection corresponding to the control connection name (`client/ctrl-connection/name`) "RouterA" presented in Section 6.1.

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <client-ip>203.0.113.1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>203.0.113.2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <state>active</state>
      </ctrl-connection>
    </server>
  </twamp>
</data>
```

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <client-ip>2001:DB8:203:0:113::1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>2001:DB8:203:0:113::2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <state>active</state>
      </ctrl-connection>
    </server>
  </twamp>
</data>
```

6.3. Session-Sender

Figure 10 shows a configuration example for a Session-Sender with session-sender/admin-state enabled, which permits a device following Figure 2 to initiate TWAMP-Test sessions.

```

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
    </session-sender>
  </twamp>
</config>

```

Figure 10: XML instance enabling Session-Sender operation.

The following configuration example shows a Session-Sender with the two TWAMP-Test sessions presented in Section 6.1.

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
      <test-session>
        <name>Test1</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <number-of-packets>900</number-of-packets>
        <periodic-interval>1</periodic-interval>
      </test-session>
      <test-session>
        <name>Test2</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <number-of-packets>900</number-of-packets>
        <lambda>1</lambda>
        <max-interval>2</max-interval>
      </test-session>
    </session-sender>
  </twamp>
</data>

```

6.4. Session-Reflector

This configuration example shows a Session-Reflector with session-reflector/admin-state enabled, which permits a device following Figure 2 to respond to TWAMP-Test sessions.

```

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
    </session-reflector>
  </twamp>
</config>

```

Figure 11: XML instance enabling Session-Reflector operation.

The following example shows the two Session-Reflector TWAMP-Test sessions corresponding to the test sessions presented in Section 6.3.

[note: '\ ' line wrapping is for formatting only]

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-\
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-\
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-\

```



```

client-ip>
  <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
  <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
  <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
  <sent-packets>21</sent-packets>
  <rcv-packets>21</rcv-packets>
  <last-sent-seq>20</last-sent-seq>
  <last-rcv-seq>20</last-rcv-seq>
</test-session>
</session-reflector>
</twamp>
</data>

```

[note: '\ ' line wrapping is for formatting only]

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>54001</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <sid>178943</sid>
      </test-session>
    </session-reflector>
  </twamp>
</data>

```

```
      <parent-connection-client-ip>203.0.113.1</parent-connection-\
client-ip>
      <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
      <parent-connection-server-ip>203.0.113.2</parent-connection-\
server-ip>
      <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
      <sent-packets>21</sent-packets>
      <rcv-packets>21</rcv-packets>
      <last-sent-seq>20</last-sent-seq>
      <last-rcv-seq>20</last-rcv-seq>
    </test-session>
  </session-reflector>
</twamp>
</data>
```

7. Security Considerations

Virtually all existing measurement systems using TWAMP [RFC5357] are administered by the same network operator. Attacks on the measurement infrastructure could be launched by third-parties to commandeer the packet generation capability, corrupt the measurements, or other examples of nefarious acts.

The YANG module specified in Section 5 of this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF [RFC6241] layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF Access Control Module (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of nodes defined in this YANG module which are writeable. These data nodes may be considered sensitive and vulnerable to attacks in some network environments. Ability to write into these nodes without proper protection can have a negative effect on the devices that support this feature.

If written, the 'admin-state' node can cause unintended test sessions to be created. If the node 'number-of-packets' that dictates how many packets are sent in any particular test session is written with

a large value, it can cause a test session to run longer than expected. Nodes that are particularly vulnerable include several timeout values put in the protocol to protect against sessions that are not active but are consuming resources. These are the REFWAIT timeout parameter which determine whether to discontinue the session if no packets are received, and nodes 'count' and 'max-count-exponent' which can cause a long time to be spent on PBKDF2 iterations. In addition, 'dscp' node marked with different DSCP markings, can cause the test traffic on the network to be skewed, and the result manipulated. Finally, nodes within 'mode-preference-chain' which specify the 'mode' and 'priority' values and indicate the preferred order of use by an operator, can be manipulated to send unauthenticated or non-encrypted traffic, enabling a MITM attack. Limiting access to these nodes will limit the ability to launch an attack in network environments.

The 'token' node defined in the model, containing a concatenation of a Challenge, AES Session-key used for encryption, and HMAC-SHA1 Session-key used for authentication, is sensitive from a privacy perspective, and can be used to disrupt a test session. The ability to read the field should be limited to the administrator of the test network.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in IETF XML Registry [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-twamp

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry YANG [RFC6020].

name: ietf-twamp

namespace: urn:ietf:params:xml:ns:yang:ietf-twamp

prefix: twamp

reference: RFC XXXX

9. Acknowledgements

We thank Fred Baker, Kevin D'Souza, Gregory Mirsky, Brian Trammell, Robert Sherman, and Marius Georgescu for their thorough and constructive reviews, comments and text suggestions.

Haoxing Shen contributed to the definition of the YANG module in Section 5.

Jan Lindblad and Ladislav Lhokta did thorough reviews of the YANG module and the examples in Appendix A.

Kostas Pentikousis was partially supported by FP7 UNIFY (<http://fp7-unify.eu>), a research project partially funded by the European Community under the Seventh Framework Program (grant agreement no. 619609). The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.

10. Contributors

Lianshu Zheng.

11. References

11.1. Normative References

- [I-D.ietf-ippm-metric-registry]
Bagnulo, M., Claise, B., Eardley, P., Morton, A., and A. Akhter, "Registry for Performance Metrics", draft-ietf-ippm-metric-registry-14 (work in progress), March 2018.
- [I-D.ietf-ippm-port-twamp-test]
Morton, A. and G. Mirsky, "OWAMP and TWAMP Well-Known Port Assignments", draft-ietf-ippm-port-twamp-test-01 (work in progress), March 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, DOI 10.17487/RFC3432, November 2002, <<https://www.rfc-editor.org/info/rfc3432>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <<https://www.rfc-editor.org/info/rfc6038>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7717] Pentikousis, K., Ed., Zhang, E., and Y. Cui, "IKEv2-Derived Shared Secret Key for the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7717, DOI 10.17487/RFC7717, December 2015, <<https://www.rfc-editor.org/info/rfc7717>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [UML] ISO/IEC, "Information technology - Open Distributed Processing - Unified Modeling Language", April 2005.

11.2. Informative References

- [NSC] John, W., Pentikousis, K., et al., "Research directions in network service chaining", Proc. SDN for Future Networks and Services (SDN4FNS), Trento, Italy IEEE, November 2013.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, DOI 10.17487/RFC5618, August 2009, <<https://www.rfc-editor.org/info/rfc5618>>.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, DOI 10.17487/RFC5938, August 2010, <<https://www.rfc-editor.org/info/rfc5938>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7312] Fabiani, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<https://www.rfc-editor.org/info/rfc7312>>.

- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <<https://www.rfc-editor.org/info/rfc8018>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

Appendix A. Detailed Data Model Examples

This appendix extends the example presented in Section 6 by configuring more fields such as authentication parameters, DSCP values and so on.

A.1. Control-Client

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <mode-preference-chain>
        <priority>0</priority>
        <mode>authenticated</mode>
      </mode-preference-chain>
      <mode-preference-chain>
        <priority>1</priority>
      </mode-preference-chain>
    </client>
  </twamp>
</data>
```

```

    <mode>unauthenticated</mode>
  </mode-preference-chain>
  <key-chain>
    <key-id>KeyClient1ToRouterA</key-id>
    <secret-key>c2VjcmV0MQ==</secret-key>
  </key-chain>
  <key-chain>
    <key-id>KeyForRouterB</key-id>
    <secret-key>c2VjcmV0Mg0K</secret-key>
  </key-chain>
  <ctrl-connection>
    <name>RouterA</name>
    <client-ip>203.0.113.1</client-ip>
    <server-ip>203.0.113.2</server-ip>
    <control-packet-dscp>32</control-packet-dscp>
    <key-id>KeyClient1ToRouterA</key-id>
    <test-session-request>
      <name>Test1</name>
      <sender-ip>203.0.113.3</sender-ip>
      <sender-udp-port>54000</sender-udp-port>
      <reflector-ip>203.0.113.4</reflector-ip>
      <reflector-udp-port>55000</reflector-udp-port>
      <padding-length>64</padding-length>
      <start-time>0</start-time>
    </test-session-request>
    <test-session-request>
      <name>Test2</name>
      <sender-ip>203.0.113.1</sender-ip>
      <sender-udp-port>54001</sender-udp-port>
      <reflector-ip>203.0.113.2</reflector-ip>
      <reflector-udp-port>55001</reflector-udp-port>
      <padding-length>128</padding-length>
      <start-time>0</start-time>
    </test-session-request>
  </ctrl-connection>
</client>
</twamp>
</data>

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <mode-preference-chain>
        <priority>0</priority>
        <mode>authenticated</mode>
      </mode-preference-chain>
    </client>
  </twamp>
</data>

```



```

<mode-preference-chain>
  <priority>1</priority>
  <mode>unauthenticated</mode>
</mode-preference-chain>
<key-chain>
  <key-id>KeyClient1ToRouterA</key-id>
  <secret-key>c2VjcmVOMQ==</secret-key>
</key-chain>
<key-chain>
  <key-id>KeyForRouterB</key-id>
  <secret-key>c2VjcmVOMg0K</secret-key>
</key-chain>
<ctrl-connection>
  <name>RouterA</name>
  <client-ip>2001:DB8:203:0:113::1</client-ip>
  <server-ip>2001:DB8:203:0:113::2</server-ip>
  <control-packet-dscp>32</control-packet-dscp>
  <key-id>KeyClient1ToRouterA</key-id>
  <test-session-request>
    <name>Test1</name>
    <sender-ip>2001:DB8:10:1:1::1</sender-ip>
    <sender-udp-port>54000</sender-udp-port>
    <reflector-ip>2001:DB8:10:1:1::2</reflector-ip>
    <reflector-udp-port>55000</reflector-udp-port>
    <padding-length>64</padding-length>
    <start-time>0</start-time>
  </test-session-request>
  <test-session-request>
    <name>Test2</name>
    <sender-ip>2001:DB8:203:0:113::1</sender-ip>
    <sender-udp-port>54001</sender-udp-port>
    <reflector-ip>2001:DB8:203:0:113::2</reflector-ip>
    <reflector-udp-port>55001</reflector-udp-port>
    <padding-length>128</padding-length>
    <start-time>0</start-time>
  </test-session-request>
</ctrl-connection>
</client>
</twamp>
</data>

```

A.2. Server

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>

```

```

    <admin-state>true</admin-state>
    <servwait>1800</servwait>
    <control-packet-dscp>32</control-packet-dscp>
    <modes>authenticated unauthenticated</modes>
    <count>15</count>
    <key-chain>
      <key-id>KeyClient1ToRouterA</key-id>
      <secret-key>c2VjcmV0MQ==</secret-key>
    </key-chain>
    <key-chain>
      <key-id>KeyClient10ToRouterA</key-id>
      <secret-key>c2VjcmV0MTANCg==</secret-key>
    </key-chain>
    <ctrl-connection>
      <client-ip>203.0.113.1</client-ip>
      <client-tcp-port>16341</client-tcp-port>
      <server-ip>203.0.113.2</server-ip>
      <server-tcp-port>862</server-tcp-port>
      <control-packet-dscp>32</control-packet-dscp>
      <selected-mode>unauthenticated</selected-mode>
      <key-id>KeyClient1ToRouterA</key-id>
      <count>15</count>
    </ctrl-connection>
  </server>
</twamp>
</data>

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <servwait>1800</servwait>
      <control-packet-dscp>32</control-packet-dscp>
      <modes>authenticated unauthenticated</modes>
      <count>15</count>
      <key-chain>
        <key-id>KeyClient1ToRouterA</key-id>
        <secret-key>c2VjcmV0MQ==</secret-key>
      </key-chain>
      <key-chain>
        <key-id>KeyClient10ToRouterA</key-id>
        <secret-key>c2VjcmV0MTANCg==</secret-key>
      </key-chain>
      <ctrl-connection>
        <client-ip>2001:DB8:203:0:113::1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>2001:DB8:203:0:113::2</server-ip>

```

```

    <server-tcp-port>862</server-tcp-port>
    <control-packet-dscp>32</control-packet-dscp>
    <selected-mode>unauthenticated</selected-mode>
    <key-id>KeyClient1ToRouterA</key-id>
    <count>15</count>
  </ctrl-connection>
</server>
</twamp>
</data>

```

A.3. Session-Sender

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
      <test-session>
        <name>Test1</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <fill-mode>zero</fill-mode>
        <number-of-packets>900</number-of-packets>
        <periodic-interval>1</periodic-interval>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <name>Test2</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <fill-mode>random</fill-mode>
        <number-of-packets>900</number-of-packets>
        <lambda>1</lambda>
        <max-interval>2</max-interval>
        <sent-packets>21</sent-packets>
        <rcv-packets>21</rcv-packets>
        <last-sent-seq>20</last-sent-seq>
        <last-rcv-seq>20</last-rcv-seq>
      </test-session>
    </session-sender>
  </twamp>
</data>

```

A.4. Session-Reflector

[note: '\ ' line wrapping is for formatting only]

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>55000</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>21</sent-packets>
        <rcv-packets>21</rcv-packets>
        <last-sent-seq>20</last-sent-seq>
        <last-rcv-seq>20</last-rcv-seq>
      </test-session>
    </session-reflector>
  </twamp>
</data>
```

```

    </test-session>
  </session-reflector>
</twamp>
</data>

```

[note: '\' line wrapping is for formatting only]

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>2001:DB8:10:1:1::1</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>2001:DB8:10:1:1::2</reflector-ip>
        <reflector-udp-port>55000</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>2001:DB8:203:0:113::1</parent-c\
onnection-client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
        <parent-connection-server-ip>2001:DB8:203:0:113::2</parent-c\
onnection-server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>2001:DB8:203:0:113::1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>2001:DB8:192:68::2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>2001:DB8:203:0:113::1</parent-c\
onnection-client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
        <parent-connection-server-ip>2001:DB8:203:0:113::2</parent-c\
onnection-server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>21</sent-packets>
      </test-session>
    </session-reflector>
  </twamp>
</data>

```

```
        <rcv-packets>21</rcv-packets>
        <last-sent-seq>20</last-sent-seq>
        <last-rcv-seq>20</last-rcv-seq>
    </test-session>
</session-reflector>
</twamp>
</data>
```

Appendix B. TWAMP Operational Commands

TWAMP operational commands could be performed programmatically or manually, e.g. using a command-line interface (CLI).

With respect to programmability, YANG can be used to define NETCONF Remote Procedure Calls (RPC), therefore it would be, in principle, possible to define TWAMP RPC operations for actions such as starting or stopping control connections or test sessions or groups of sessions; retrieving results; clearing stored results, and so on.

However, TWAMP [RFC5357] does not attempt to describe such operational actions. Refer also to Section 2 and the unlabeled links in Figure 1. In actual deployments different TWAMP implementations may support different sets of operational commands, with different restrictions. Therefore, this document considers it the responsibility of the individual implementation to define its corresponding TWAMP operational commands data model.

Authors' Addresses

Ruth Civil
Ciena Corporation
307 Legget Drive
Kanata, ON K2K 3C8
Canada

Email: gcivil@ciena.com
URI: www.ciena.com

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

Reshad Rahman
Cisco Systems
2000 Innovation Drive
Kanata, ON K2K 3E8
Canada

Email: rrahman@cisco.com

Mahesh Jethanandani
Xoriant Corporation
1248 Reamswood Drive
Sunnyvale, CA 94089
USA

Email: mjethanandani@gmail.com

Kostas Pentikousis (editor)
Travelping
Siemensdamm 50
Berlin 13629
Germany

Email: k.pentikousis@travelping.com

Network Working Group
Internet-Draft
Updates: 5357 (if approved)
Intended status: Standards Track
Expires: December 2, 2017

G. Mirsky
ZTE Corp.
M. Perumal
Ericsson
R. Foote
Nokia
L. M. Contreras
Telefonica
L. Jalil
Verizon
May 31, 2017

UDP Port Allocation for the Receiver Port in Two-Way Active Measurement
Protocol (TWAMP)
draft-mirsky-ippm-twamp-refl-registered-port-03

Abstract

This document arguments and requests re-allocation of an UDP port number from the System Ports range for a Reflector in Two-Way Active Measurement Protocol (TWAMP). This document, if accepted, will be an update to the TWAMP Test protocol specified in RFC 5357.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 2, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Requirements Language	2
3. Impact to TWAMP-Control Protocol	3
4. Impact to TWAMP-Test Protocol	3
5. IANA Considerations	4
6. Security Considerations	4
7. Acknowledgments	5
8. Normative References	5
Authors' Addresses	5

1. Introduction

One particular compelling vision of the Two-Way Active Measurement Protocol (TWAMP) [RFC5357] is widespread deployment of open servers that would make IP Performance Metrics (IPPM) measurements a commonplace. This is complemented by the proliferation of the Internet of Things (IoT) devices, such as sensors, and the need for obtaining IPPM measurements from those devices by the service provider. IoT devices are often constrained by limited processing power and memory and benefit from TWAMP Light, as defined in Appendix I [RFC5357].

TWAMP Light provides a simple solution for devices to act as test points in the network, by avoiding the need for the TWAMP-Control protocol [RFC5357]. In the absence of TWAMP-Control, a registered (default) UDP port that can be used as the Receiver Port for TWAMP-Test will simplify configuration and management of the TWAMP-Light test sessions.

This document requests re-allocation of the UDP port number from the System Ports range [RFC6335] as Receiver Port for TWAMP-Test.

2. Requirements Language

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

3. Impact to TWAMP-Control Protocol

Section 3.5 [RFC5357] describes in details the process of negotiating value of the Receiver Port. The Control-Client, acting on behalf of the Session-Sender, proposes the port number from the Dynamic Port range [RFC6335]:

"The Receiver Port is the desired UDP port to which TWAMP-Test packets will be sent by the Session-Sender (the port where the Session-Reflector is asked to receive test packets). The Receiver Port is also the UDP port from which TWAMP-Test packets will be sent by the Session-Reflector (the Session-Reflector will use the same UDP port to send and receive packets)."

But the proposed Receiver Port may be not available, e.g. being in use by other test session or another application. In this case:

"... the Server at the Session-Reflector MAY suggest an alternate and available port for this session in the Port field. The Session-Sender either accepts the alternate port, or composes a new Session-Request message with suitable parameters. Otherwise, the Server uses the Accept field to convey other forms of session rejection or failure to the Control Client and MUST NOT suggest an alternate port; in this case, the Port field MUST be set to zero."

The allocated TWAMP Receiver Port number Section 5 MAY be advertised by the Server. The Control Client that supports use of the allocated TWAMP Receiver Port MUST accept the port number advertised by the Server. If the Server does not support the allocated TWAMP Receiver Port, then it sends another Session-Request message with new parameters. Thus the deployment of the allocated TWAMP Receiver Port number is backward compatible with existing TWAMP-Control solutions that are based on [RFC5357]. At the same time, use of the UDP port number allocated from the User Port range [RFC6335] will help to avoid the situation when the Server finds the proposed port being already in use.

4. Impact to TWAMP-Test Protocol

TWAMP-Test may be used to measure IP performance metrics in an Equal Cost Multipath (ECMP) environment. Though algorithms to balance IP flows among available paths had not been standardized, the most common is the Five-tuple that uses destination IP address, source IP address, protocol type, destination port number, and source port number. To attempt to monitor different paths in ECMP network is sufficient to variate only one of five parameters, e.g. the source port number. Thus, there will be no negative impact on ability to have concurrent TWAMP test sessions between the same test points to

monitor different paths in the ECMP network when using the allocated UDP port number as the Receiver Port.

The allocation of the TWAMP Receiver Port from the User Port Range [RFC6335] benefits TWAMP Light mode of the TWAMP-Test. The allocated UDP port number Section 5 may be used as default value for the Receiver Port to simplify configuration and management of the TWAMP-Light test sessions.

5. IANA Considerations

The Service Name and Transport Protocol Port Number Registry defined in [RFC6335].

[RFC5357] has been allocated UDP port 862 for TWAMP-Control protocol. IANA is requested to re-assign UDP port 862 as follows:

Service Name	Port Number	Transport Protocol	Description	Semantics Definition	Reference
twamp-test	862	UDP	TWAMP-Test Receiver Port	Section 4	This document

Table 1: TWAMP Receiver Port

6. Security Considerations

The registered UDP port as the Receiver Port for TWAMP-Test may be used as target of denial-of-service (DoS) or used by man-in-the-middle (MitM) attack. To improve protection from the DoS following methods are recommended:

- o filtering access to the TWAMP Receiver Port by access list;
- o non-routable IPs outside of the domain for the TWAMP loopback.

MitM attack may try to modify the content of the TWAMP-Test packet thus altering measurement results. An implementation can use data consistency check to detect modification of data. In addition, it can use encryption of TWAMP-Test packets to prevent eavesdropping.

7. Acknowledgments

TBD

8. Normative References

- [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>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<http://www.rfc-editor.org/info/rfc5357>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<http://www.rfc-editor.org/info/rfc6335>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Muthu Arul Mozhi Perumal
Ericsson
Ferns Icon
Doddanekundi, Mahadevapura
Bangalore, Karnataka 560037
India

Email: p.muthu.arul.mozhi@ericsson.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

Luis M. Contreras
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 31, 2017

T. Mizrahi
C. Arad
Marvell
G. Fioccola
M. Cociglio
Telecom Italia
M. Chen
L. Zheng
Huawei Technologies
G. Mirsky
ZTE Corp.
June 29, 2017

Compact Alternate Marking Methods for Passive Performance Monitoring
draft-mizrahi-ippm-multiplexed-alternate-marking-02

Abstract

This memo introduces new alternate marking methods that require a compact overhead of either a single bit per packet, or zero bits per packet. This memo also presents a summary of alternate marking methods, and discusses the tradeoffs among them. The target audience of this document is network protocol designers; this document is intended to help protocol designers choose the best alternate marking method(s) based on the protocol's constraints and requirements.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 31, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Background	3
1.2. The Scope of This Document	4
2. Terminology	5
2.1. Requirements Language	5
2.2. Abbreviations	5
3. Marking Abstractions	5
4. Double Marking	7
5. Single-bit Marking	8
5.1. Single Marking Using the First Packet	8
5.2. Single Marking using the Mean Delay	8
5.3. Alternate Marking using a Multiplexed Marking Bit	8
5.3.1. Overview	8
5.3.2. Timing and Synchronization Aspects	9
5.4. Pulse Marking	11
6. Zero-bit Marking	12
6.1. Hash-based Sampling	12
6.2. Hashed Pulse Marking	13
6.3. Hashed Double Marking	13
6.4. Mixed Hashed Marking	14
7. Summary of Marking Methods	14
8. Alternate Marking using Reserved Values	17
9. IANA Considerations	18
10. Security Considerations	18
11. References	18
11.1. Normative References	18
11.2. Informative References	19
Authors' Addresses	20

1. Introduction

1.1. Background

Alternate marking, defined in [I-D.ietf-ippm-alt-mark], is a method for measuring packet loss, packet delay, and packet delay variation. Typical delay measurement protocols require the two measurement points (MPs) to exchange timestamped test packets. In contrast, the alternate marking method does not require control packets to be exchanged. Instead, every data packet carries a color indicator, which divides the traffic into consecutive blocks of packets.

The color value is toggled periodically, as illustrated in Figure 1.

A: packet with color 0
 B: packet with color 1

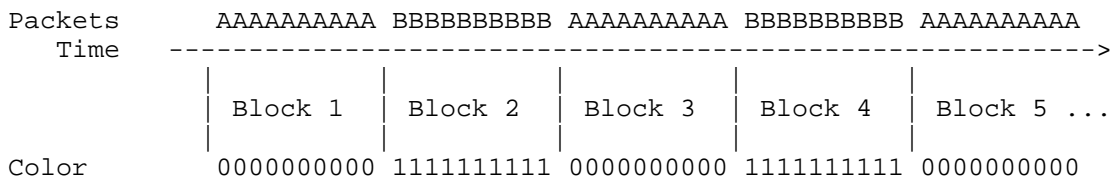


Figure 1: Alternate marking: packets are monitored on a per-color basis.

Alternate marking is used between two MPs, the initiating MP, and the monitoring MP. The initiating MP incorporates the marking field into en-route packets, allowing the monitoring MP to use the marking field in order to bind each packet to the corresponding block.

Each of the MPs maintains two counters, one per color. At the end of each block the counter values can be collected by a central management system, and analyzed; the packet loss can be computed by comparing the counter values of the two MPs.

When using alternate marking delay measurement can be performed in one of three ways (as per [I-D.ietf-ippm-alt-mark]):

- o Single marking using the first packet: in this method each packet uses a single marking bit, used as a color indicator. The first packet of each block is used by both MPs as a reference for delay measurement. The timestamp of this packet is measured by the two measurement points, and can be collected by the mangement system from each of the measurement points, which can compute the path delay by comparing the two timestamps. The drawback of this

approach is that it is not accurate when packets arrive out-of-order, as the two MPs may have a different view of which packet was the first in the block.

- o Single marking using the mean delay: as in the previous method, each packet uses a single marking method, indicating the color. Each of the MPs computes the average packet timestamp of each block. The management system can then compute the delay by comparing the average times of the two MPs. The drawback of this approach is that it may be computationally heavy, or difficult to implement at the data plane.
- o Double marking: each packet uses two marking bits. One bit is used as a color indicator, and one is used as a timestamping indicator. This method resolves the drawbacks raised for the two previous methods, at the expense of an extra bit in the packet header.

The double marking method is the most straightforward approach. It allows for accurate measurement without incurring expensive computational load. However, in some cases allocating two bits for passive measurement is not possible. For example, if alternate marking is implemented over IPv4, allocating 2 marking bits in the IPv4 header is challenging, as every bit in the 20-octet header is costly; one of the possible approaches discussed in [I-D.ietf-ippm-alt-mark] is to reserve one or two bits from the DSCP field for remarking. In this case every marking bit comes at the expense of reducing the DSCP range by a factor of two.

1.2. The Scope of This Document

This memo extends the marking methods of [I-D.ietf-ippm-alt-mark], and introduces methods that require a single marking bit, or zero marking bits.

Two single-bit marking methods are proposed, multiplexed marking and pulse marking. In multiplexed marking the color indicator and the timestamp indicator are multiplexed into a single bit, providing the advantages of the double marking method while using a single bit in the packet header. In pulse marking both delay and loss measurement are triggered by a 'pulse' value in a single marking field.

This document also discusses zero-bit marking methods that leverage well-known hash-based selection [RFC5475] approaches.

Alternate marking is discussed in this memo as a single-bit or a two-bit marking method. However, these methods can similarly be applied to larger fields, such as an IPv6 Flow Label or an MPLS Label;

single-bit marking can be applied using two reserved values, and two-bit marking can be applied using four reserved values. Marking based on reserved values is further discussed in this document, including its application to MPLS and IPv6.

Finally, this memo summarizes the alternate marking methods, and discusses the tradeoffs among them. It is expected that different network protocols will have different constraints, and therefore may choose to use different alternate marking methods. In some cases it may be preferable to support more than one marking method; in this case the particular marking method may be signaled through the control plane.

2. Terminology

2.1. Requirements Language

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

2.2. Abbreviations

The following abbreviations are used in this document:

DSCP	Differentiated Services Code Point
DM	Delay Measurement
LM	Loss Measurement
LSP	Label Switched Path
MP	Measurement Point
MPLS	Multiprotocol Label Switching
SFL	Synonymous Flow Label [I-D.bryant-mpls-sfl-framework]

3. Marking Abstractions

The marking methods that were discussed in Section 1, as well as the methods introduced in this document, use two basic abstractions, pulse detection, and step detection.

The common thread along the various marking methods is that one or two marking bits are used by the MPs to signal a measurement event.

The value of the marking bit indicates when the event takes place, in one of two ways:

- Pulse An event is detected when the value of the marking bit is toggled in a single packet.
- Step An event is detected when the value of the marking bit is toggled, and remains at the new value.

The double marking method (Section 1) uses pulse-based detection for DM, and step-based detection for LM.

Pulse-based detection affects the processing of a single packet; the packet that indicates the pulse is processed differently than the packets around it. For example, in the double marking method, the marked packet is timestamped for DM, without affecting the packets before or after it. Note that if the marked packet is lost, no pulse is detected, yielding a missing measurement (see Figure 2).

P: indicates a packet

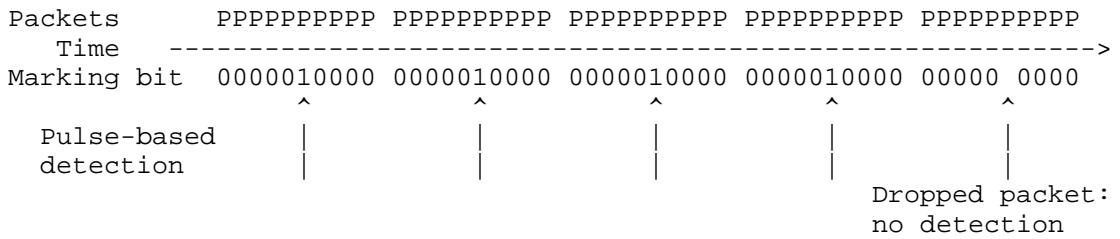


Figure 2: Pulse-based Detection.

In step-based detection the event is detected by observing a value change in stream of packets. Specifically, when the step approach is used for LM (as in the double marking method), two counters are used per flow; each MP decides which counter to use based on the value of the marking bit. Thus, the step-based approach allows accurate counting even when packets arrive out-of-order (see Figure 3). When the step approach is used for DM (e.g., single marking using the first packet), out-of-order causes the delay measurement to be false, without any indication to the management system.

5. Single-bit Marking

5.1. Single Marking Using the First Packet

This method uses a single marking bit that indicates the color, as described in [I-D.ietf-ippm-alt-mark]. Both LM and DM are implemented using a step-based approach; LM is implemented using two color-based counters per flow. The first packet of every period is used by the two MPs as the reference for measuring the delay. As denoted above, the delay computed in this method may be erroneous when packets are delivered out-of-order.

A: packet with color 0
 B: packet with color 1

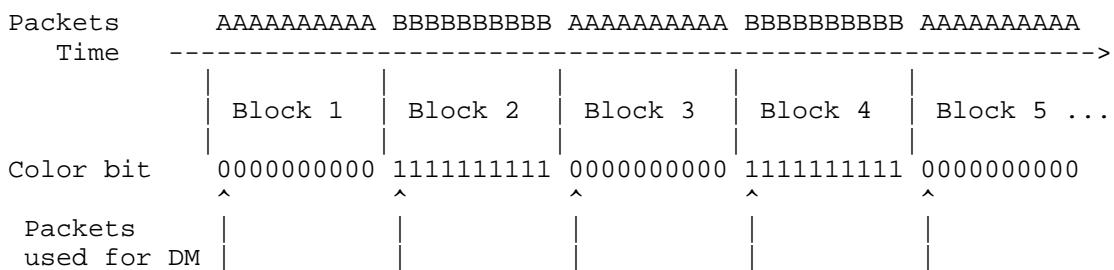


Figure 5: Single marking using the first packet of the block.

5.2. Single Marking using the Mean Delay

As in the first-packet approach, in the mean delay approach ([I-D.ietf-ippm-alt-mark]) a single marking bit is used to indicate the color, enabling step-based loss measurement. Delay is measured in each period by averaging the measured delay over all the packets in the period. As discussed above, this approach is not sensitive to out-of-order delivery, but may be heavy from a computational perspective.

5.3. Alternate Marking using a Multiplexed Marking Bit

5.3.1. Overview

This section introduces a method that uses a single marking bit that serves two purposes: a color indicator, and a timestamp indicator. The double marking method that was discussed in the previous section uses two 1-bit values: a color indicator C, and a timestamp indicator

T. The multiplexed marking bit, denoted by M, is an exclusive or between these two values: $M = C \text{ XOR } T$.

An example of the use of the multiplexed marking bit is depicted in Figure 6. The example considers two routers, R1 and R2, that use the multiplexed bit method to measure traffic from R1 to R2. In each block R1 designates one of the packets for delay measurement. In each of these designated packets the value of the multiplexed bit is reversed compared to the other packets in the same block, allowing R2 to distinguish the designated packets from the other packets.

A: packet with color 0
 B: packet with color 1

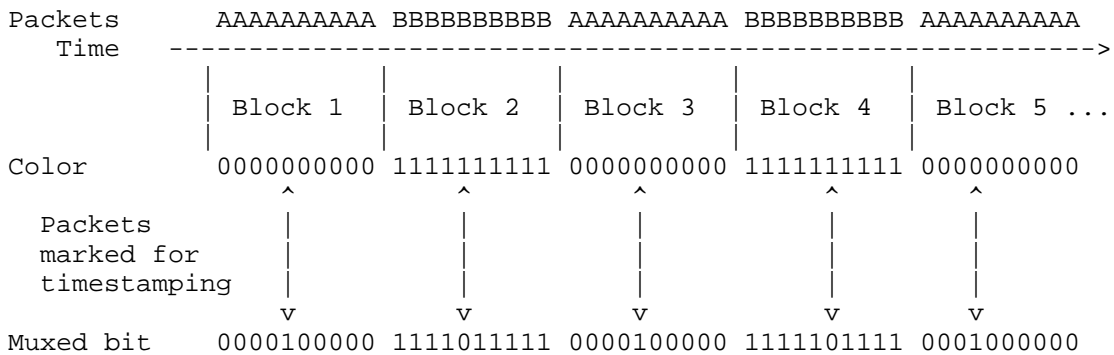


Figure 6: Alternate marking with multiplexed bit.

5.3.2. Timing and Synchronization Aspects

It is assumed that all MPs are synchronized to a common reference time with an accuracy of $\pm A/2$. Thus, the difference between the clock values of any two MPs is bounded by A. Clocks can be synchronized for example using NTP [RFC5905], PTP [IEEE1588], or by other means. The common reference time is used for dividing the time domain into equal-sized measurement periods, such that all packets forwarded during a measurement period have the same color, and consecutive periods have alternating colors.

The single marking bit incorporates two multiplexed values. From the monitoring MP's perspective, the two values are Time-Division Multiplexed (TDM), as depicted in Figure 7. It is assumed that the start time of every measurement period is known to both the initiating MP and the monitoring MP. If the measurement period is L, then during the first and the last L/4 time units of each block the

marking bit is interpreted by the monitoring MP as a color indicator. During the middle part of the block, the marking bit is interpreted as a timestamp indicator; if the value of this bit is different than the color value, the corresponding packet is used as a reference for delay measurement.

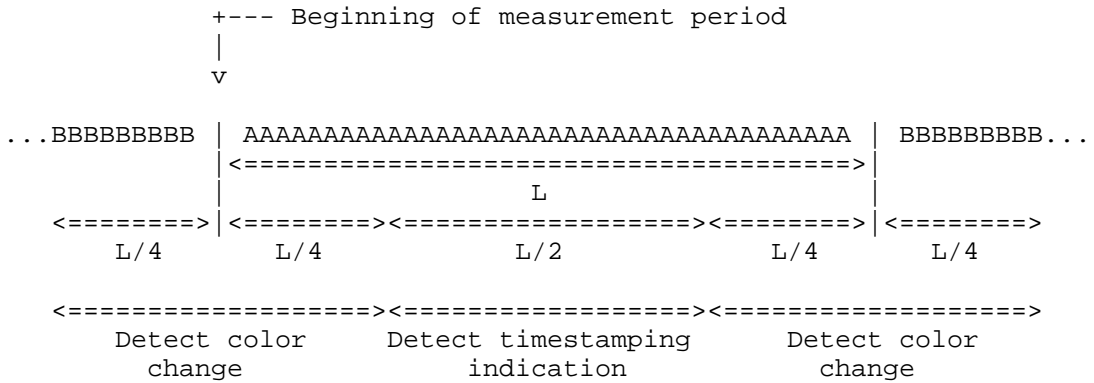


Figure 7: Multiplexed marking field interpretation at the receiving measurement point.

In order to prevent ambiguity in the receiver's interpretation of the marking field, the initiating MP is permitted to set the timestamp indication only during a specific interval, as depicted in Figure 8. Since the receiver is willing to receive the timestamp indication during the middle L/2 time units of the block, the sender refrains from sending the timestamp indication during a guardband interval of d time units at the beginning and end of the L/2-period.

- o The value of the counter is captured for LM.

In each period, each of the MPs exports the timestamp and counter-stamp to the management system, which can then compute the loss and delay in that period. It should be noted that as in [I-D.ietf-ippm-alt-mark], if the length of the measurement period is L time units, then all network devices must be synchronized to the same clock reference with an accuracy of $\pm L/2$ time units.

The pulse marking approach is illustrated in Figure 9. Since both LM and DM use a pulse-based trigger, if the marked packet is lost then no measurement is available in this period. Moreover, the LM accuracy may be affected by out-of-order delivery.

P: packet - all packets have the same color

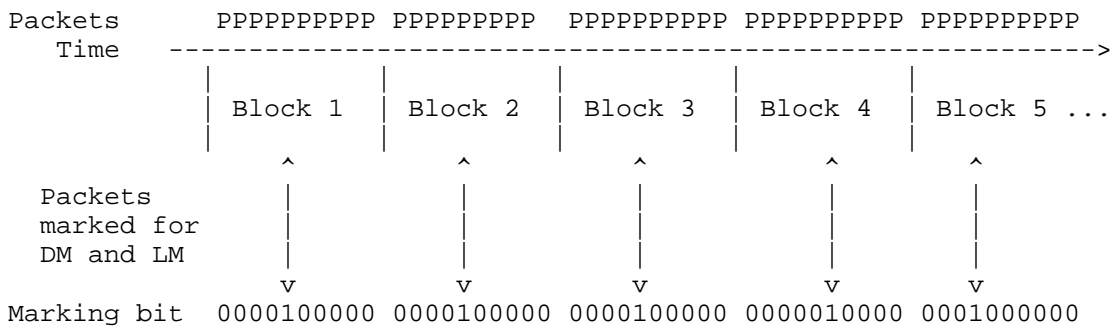


Figure 9: Pulse marking method.

6. Zero-bit Marking

6.1. Hash-based Sampling

Hash based selection [RFC5475] is a well-known method for sampling a subset of packets. As defined in [RFC5475]:

A Hash Function h maps the Packet Content c , or some portion of it, onto a Hash Range R . The packet is selected if $h(c)$ is an element of S , which is a subset of R called the Hash Selection Range.

Hash-based selection can be leveraged as a marking method, allowing a zero-bit marking approach. Specifically, the pulse and step abstractions can be implemented using hashed selection:

- o Hashed pulse-based trigger: in this approach, a packet is selected if $h(c)$ is an element of S , which is a strict subset of the hash range R . When $|S| \ll |R|$, the average sampling period is long, reducing the probability of ambiguity between consecutive packets. $|S|$ and $|R|$ denote the number of elements in S and R , respectively.
- o Hashed step-based trigger: the hash values of a given traffic flow are said to be monotonically increasing if for two packets p_1 and p_2 , if p_1 is sent before p_2 then $h(p_1) \leq h(p_2)$. If it is guaranteed that the hash values of a flow are monotonically increasing, then a step-based approach can be used on the range R . For example, in an IPv4 flow the Identification field can be used as the hash value of each packet. Since the Identification field is monotonically increasing, the step-based trigger can be implemented using consecutive ranges of the Identification value. For example, the fourth bit of the Identification field is toggled every 8 packets. Thus, a possible hash function simply takes the fourth bit of the Identification field as the hash value. This hash value is toggled every 8 packets, simulating the alternate marking behavior of Section 4.

Note that as opposed to the double marking and single marking methods, hashed sampling is not based on fixed time intervals, as the duration between sampled packets depends only on the hash value.

It is also important to note that all methods that use hash-based marking require the hash function and the set S to be configured consistently across the MPs.

6.2. Hashed Pulse Marking

In this approach a hash is computed over the packet content, and both LM and DM are triggered based on the pulse-based trigger (Section 6.1). A pulse is detected when the hash value $h(c)$ is equal to one of the values in S . The hash function h and the set S determine the probability (or frequency) of the pulse event.

6.3. Hashed Double Marking

As in the previous approach, hashed double marking also uses a hash that is computed over the packet content. In this approach DM is performed using a pulse-based trigger, whereas the LM trigger is step-based (Section 6.1). The main drawback of this method is that the step-based trigger is possible only under the assumption that the hash function is monotonically increasing, which is not necessarily possible in all cases. Specifically, a measured flow is not necessarily an IPv4 5-tuple. For example, a measured flow may

include multiple IPv4 5-tuple flows, and in this case the Identification field is not monotonically increasing.

6.4. Mixed Hashed Marking

Mixed hashed marking combines the single marking approach with hash-based sampling. A single marking bit is used in the packet header as a color indicator, while a hash-based pulse is used to trigger DM. Although this method requires a single bit, it is described in this section as it is closely related to the other hash-based methods that require zero marking bits.

7. Summary of Marking Methods

This section summarizes the marking methods described in this memo. Each row in the table of Figure 10 represents a marking method. For each method the table specifies the number of bits required in the header, the number of counters per flow for LM, the methods used for LM and DM (pulse or step), and also the resilience to disturbances.

Method	# of bits	# of counters	LM Method	DM Method	Resilience to Reordering		Resilience to packet drops	
					LM	DM	LM	DM
Double marking	2	2	Step	Pulse	+	+	+	-
Single marking - 1st packet	1	2	Step	Step	+	--	+	--
Single marking - mean delay	1	2	Step	Mean	+	+	+	-
Multiplexed marking	1	2	Step	Pulse	+	+	+	-
Pulse marking	1	1	Pulse	Pulse	--	+	-	-
Hashed pulse marking	0	1	Hashed pulse	Hashed pulse	--	+	-	-
Hashed double marking	0	2	Hashed step*	Hashed pulse	+	+	+	-
Mixed hashed marking	1	2	Step	Hashed pulse	+	+	+	-

- + Accurate measurement.
- No measurement in case of disturbance (detectable).
- False measurement in case of disturbance (not detectable).
- * Hashed step works only when the hash is monotonically increasing.

Figure 10: Summary of Marking Methods

In the context of this comparison two possible disturbances are considered: out-of-order delivery, and packet drops. Generally speaking, pulse based methods are sensitive to packet drops, since if the marked packet is dropped no measurement is recorded in the current period. Notably, a missing measurement is detectable by the management system, and is not as severe as a false measurement. Step-based triggers are generally resilient to out-of-order delivery for LM, but are not resilient to out-of-order delivery for DM. Notably, a step-based trigger may yield a false delay measurement when packets are delivered out-of-order, and this inaccuracy is not detectable.

As mentioned above, the double marking method is the most straightforward approach, and is resilient to most of the disturbances that were analyzed. Its obvious drawback is that it requires two marking bits.

Several single marking methods are discussed in this memo. In this case there is no clear verdict which method is the optimal one. The first packet method may be simple to implement, but may present erroneous delay measurements in case of dropped or reordered packets. Arguably, the mean delay approach and the multiplexed approach may be more difficult to implement (depending on the underlying platform), but are more resilient to the disturbances that were considered here. Note that the computational complexity of the mean delay approach can be reduced by combining it with a hashed approach, i.e., by computing the mean delay over a hash-based subset of the packets. The pulse marking method requires only a single counter per flow, while the other methods require two counters per flow.

The hash-based sampling approaches reduce the overhead to zero bits, which is a significant advantage. However, the sampling period in these approaches is not associated with a fixed time interval. Therefore, in some cases adjacent packets may be selected for the sampling, potentially causing measurement errors. Furthermore, when the traffic rate is low, measurements may become significantly infrequent.

It should be noted that most of the marking methods that were presented in this memo are intended for point-to-point measurements, e.g., from MP1 to MP2 in Figure 11. In point-to-multipoint measurements, the mean delay method can be used to measure the loss and delay of the entire point-to-multipoint flow (which includes all the traffic from MP3 to either MP4 or MP5), while other methods such as double marking can be used to measure the point-to-point performance, for example from MP3 to MP5. Alternate marking in multipoint scenarios is discussed in detail in [I-D.fioccola-ippm-multipoint-alt-mark].

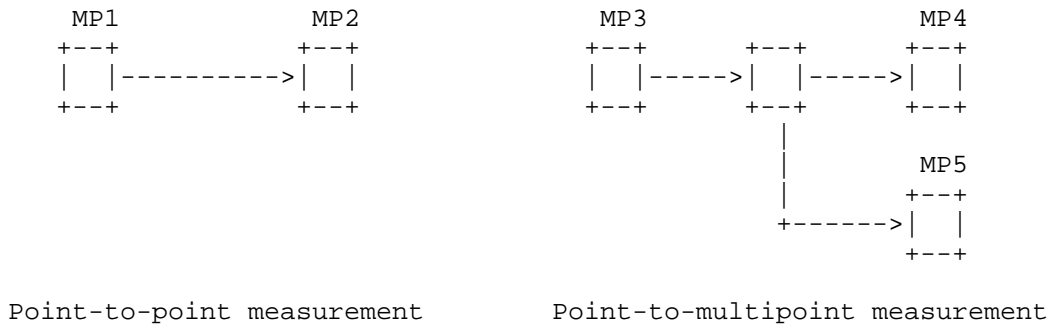


Figure 11: Point-to-point and point-to-multipoint measurements.

8. Alternate Marking using Reserved Values

As mentioned in Section 1, a marking bit is not necessarily a single bit, but may be implemented by using two well-known values in one of the header fields. Similarly, two-bit marking can be implemented using four reserved values.

A notable example is MPLS Synonymous Flow Labels (SFL), as defined in [I-D.bryant-mpls-rfc6374-sfl]. Two MPLS Label values can be used to indicate the two colors of a given LSP: the original Label value, and an SFL value. A similar approach can be applied to IPv6 using the Flow Label field.

The following example illustrates how alternate marking can be implemented using reserved values. The bit multiplexing approach of Section 5.3 is applicable not only to single-bit color indicators, but also to two-value indicators; instead of using a single bit that is toggled between '0' and '1', two values of the indicator field, U and W, can be used in the same manner, allowing both loss and delay measurement to be performed using only two reserved values. Thus, the multiplexing approach of Figure 6 can be illustrated more generally with two values, U and W, as depicted in Figure 12.

A: packet with color 0
 B: packet with color 1

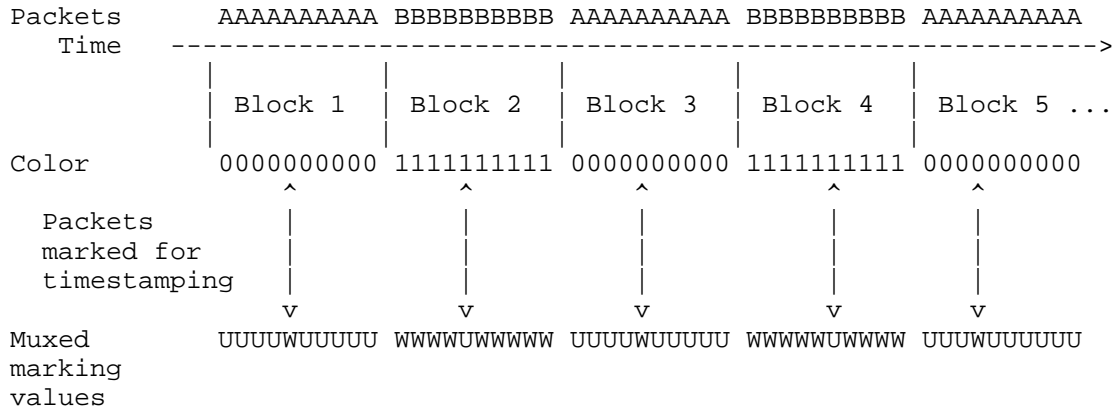


Figure 12: Alternate marking with two multiplexed marking values, U and W.

9. IANA Considerations

This memo includes no requests from IANA.

10. Security Considerations

The security considerations of the alternate marking method are discussed in [I-D.ietf-ippm-alt-mark]. The analysis of Section 7 emphasizes the sensitivity of some of the alternate marking methods to packet drops and to packet reordering. Thus, a malicious attacker may attempt to tamper with the measurements by either selectively dropping packets, or by selectively reordering specific packets. The multiplexed marking method Section 5.3 that is defined in this document requires slightly more stringent synchronization than the conventional marking method, potentially making the method more vulnerable to attacks on the time synchronization protocol. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

11. References

11.1. Normative References

- [I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L.,
Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
"Alternate Marking method for passive and hybrid
performance monitoring", draft-ietf-ippm-alt-mark-05 (work
in progress), June 2017.
- [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>>.

11.2. Informative References

- [I-D.bryant-mpls-rfc6374-sfl]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,
Mirsky, G., and G. Fioccola, "RFC6374 Synonymous Flow
Labels", draft-bryant-mpls-rfc6374-sfl-04 (work in
progress), April 2017.
- [I-D.bryant-mpls-sfl-framework]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,
and G. Mirsky, "Synonymous Flow Label Framework", draft-
bryant-mpls-sfl-framework-05 (work in progress), June
2017.
- [I-D.fioccola-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto,
"Multipoint Alternate Marking method for passive and
hybrid performance monitoring", draft-fioccola-ippm-
multipoint-alt-mark-00 (work in progress), June 2017.
- [IEEE1588]
IEEE, "IEEE 1588 Standard for a Precision Clock
Synchronization Protocol for Networked Measurement and
Control Systems Version 2", 2008.
- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F.
Raspall, "Sampling and Filtering Techniques for IP Packet
Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009,
<<http://www.rfc-editor.org/info/rfc5475>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
"Network Time Protocol Version 4: Protocol and Algorithms
Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
<<http://www.rfc-editor.org/info/rfc5905>>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<http://www.rfc-editor.org/info/rfc7384>>.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Carmi Arad
Marvell
6 Hamada st.
Yokneam
Israel

Email: carmi@marvell.com

Giuseppe Fioccola
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Mach(Guoyi) Chen
Huawei Technologies

Email: mach.chen@huawei.com

Lianshu Zheng
Huawei Technologies

Email: vero.zheng@huawei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com