

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 18, 2016

M. Chen, Ed.
L. Zheng, Ed.
Huawei Technologies
G. Mirsky, Ed.
Ericsson
G. Fioccola, Ed.
Telecom Italia
T. Mizrahi, Ed.
Marvell
March 17, 2016

IP Flow Performance Measurement Framework
draft-chen-ippm-coloring-based-ipfpm-framework-06

Abstract

This document specifies a measurement method, the IP flow performance measurement (IPFPM). With IPFPM, data packets are marked into different blocks of markers by changing one or more bits of packets. No additional delimiting packet is needed and the performance is measured in-service and in-band without the insertion of additional traffic.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 18, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	4
3. Overview and Concept	4
4. Consideration on Marking Bits	6
5. Reference Model and Functional Components	6
5.1. Reference Model	6
5.2. Measurement Control Point	7
5.3. Measurement Agent	7
6. Period Number	8
7. Re-ordering Tolerance	8
8. Packet Loss Measurement	9
9. Packet Delay Measurement	10
10. Synchronization Aspects	11
10.1. Synchronization for the Period Number	12
10.2. Synchronization for Delay Measurement	12
11. IANA Considerations	13
12. Security Considerations	13
13. Acknowledgements	14
14. Contributing Authors	14
15. References	15
15.1. Normative References	15
15.2. Informative References	15
Authors' Addresses	17

1. Introduction

Performance Measurement (PM) is an important tool for service providers, used for Service Level Agreement (SLA) verification, troubleshooting (e.g., fault localization or fault delimitation) and network visualization. Measurement methods could be roughly put into two categories - active measurement methods and passive measurement

methods. Active methods measure performance or reliability parameters by the examination of traffic (IP Packets) injected into the network, expressly for the purpose of measurement by the intended measurement points. In contrast, passive method measures some performance or reliability parameters associated with the existing traffic (packets) on the network. Both passive and active methods have their strengths and should be regarded as complementary. There are certain scenarios where active measurement alone is not enough or applicable and passive measurement is desirable[I-D.deng-ippm-passive-wireless-usecase].

With active measurement methods, the rate, numbers and interval between the injected packets may affect the accuracy of the results. Moreover, injected test packets are not always guaranteed to be in-band with the data traffic in the pure IP network due to Equal Cost Multi-Path (ECMP).

The Multiprotocol Label Switching (MPLS) PM protocol [RFC6374] for packet loss could be considered an example of a passive performance measurement method. By periodically inserting auxiliary Operations, Administration and Maintenance (OAM) packets, the traffic is delimited by OAM packets into consecutive blocks, and the receivers count the packets and calculate the packets lost in each block. However, solutions like [RFC6374] depend on the fixed positions of the delimiting OAM packets for packets counting, and thus are vulnerable to out-of-order arrival of packets. This could happen particularly with out-of-band OAM channels, but might also happen with in-band OAM because of the presence of multipath forwarding within the network. Out of order delivery of data and the delimiting OAM packets can give rise to inaccuracies in the performance measurement figures. The scale of these inaccuracies will depend on data speeds and the variation in delivery, but with out-of-band OAM, this could result in significant differences between real and reported performance.

This document specifies a different measurement method, the IP flow performance measurement (IPFPM). With IPFPM, data packets are marked into different blocks of markers by changing one or more bits of packets without altering normal processing in the network. No additional delimiting packet is needed and the performance can be measured in-service without the insertion of additional traffic. Furthermore, because marking-based IP performance measurement does not require extra OAM packets for traffic delimitation, it can be used in situations where there is packet re-ordering. IP Flow Information eXport (IPFIX) [RFC7011] is used for reporting the measurement data of IPFPM to a central calculation element for performance metrics calculation. Several new Information Elements of

IPFIX are defined for IPFPM. These are described in the companion document [I-D.chen-ippm-ipfpm-report].

2. Terminology

The acronyms used in this document will be listed here.

3. Overview and Concept

The concept of marking IP packets for performance measurement is described in [I-D.tempia-opsawg-p3m]. Marking of packets in a specific IP flow to different colors divides the flows into different consecutive blocks. Packets in a block have same marking and consecutive blocks will have different markings. This enables the measuring node to count and calculate packet loss and/or delay based on each block of markers without any additional auxiliary OAM packets. The following figure (Figure 1) is an example that illustrates the different markings in a single IP flow in alternate 0 and 1 blocks.

```
| 0 Block | 1 Block | 0 Block | 1 Block |
000000000000 111111111111 000000000000 111111111111
```

Figure 1: Packet Marking

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

For packet delay measurement, there are three solutions. One is similar to the packet loss, it still marks the IP flows to different blocks of markers and uses the time of the marking change as the reference time for delay calculations. This solution requires that there must not be any out-of-order packets; otherwise, the result will not be accurate. Because it uses the first packet of each block of markers for delay measurement, if there is packet reordering, the first packet of each block at the sender will be probably different from the first packet of the block at the receiver. An alternate way is to periodically mark a single packet in the IP flow. Within a given time period, there is only one packet that can be marked. The sender records the timestamp when the marked packet is transmitted, and the receiver records the timestamp when receiving the marked packet. With the two timestamps, the packet delay can be computed.

An additional method consists of taking into account the average arrival time of the packets within a single block (i.e. the same block of markers used for packet loss measurement). The network device locally sums all the timestamps and divides by the total number of packets received, so 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 average delay between those nodes. This method is robust to out of order packets and also to packet loss (only an error is introduced dependent from the number of lost packets).

A centralized calculation element Measurement Control Point (MCP) is introduced in Section 5.2 of this document, to collect the packet counts and timestamps from the senders and receivers for metrics calculation. The IP Flow Information eXport (IPFIX) [RFC7011] protocol is used for collecting the performance measurement statistic information [I-D.chen-ippm-ipfpm-report]. For the statistic information collected, the MCP has to know exactly what packet pair counts (one from the sender and the other is from the receiver) are based on the same block of markers and a pair of timestamps (one from the sender and the other is from the receiver) are based on the same marked packet. In case of average delay calculation the MCP has to know in addition to the packet pair counters also the pair of average timestamps for the same block of markers. The "Period Number" based solution Section 6 is introduced to achieve this.

For a specific IP flow to be measured, there may be one or more upstream and downstream Measurement Agents (MAs) (Section 5.3). An IP flow can be identified by the Source IP (SIP) and Destination IP (DIP) addresses, and it may combine the SIP and DIP with any or all of the Protocol number, the Source port, the Destination port, and the Type of Service (TOS) to identify an IP flow. For each flow, there will be a flow identifier that is unique within a certain administrative domain. To simplify the process description, the flows discussed in this document are all unidirectional. A bidirectional flow can be seen as two unidirectional flows.

IPPFM supports the measurement of a Multipoint-to-Multipoint (MP2MP) model, which satisfies all the scenarios that include Point-to-Point (P2P), Point-to-Multipoint (P2MP), Multipoint-to-Point (MP2P), and MP2MP. The P2P scenario is obvious and can be used anywhere. P2MP and MP2P are very common in mobile backhaul networks. For example, a Cell Site Gateway (CSG) that uses multi-homing to two Radio Network Controller (RNC) Site Gateways (RSGs) is a typical network design. When there is a failure, there is a requirement to monitor the flows between the CSG and the two RSGs hence to determine whether the fault is in the transport network or in the wireless network (typically called "fault delimitation"). This is especially useful in the

situation where the transport network belongs to one service provider and the wireless network belongs to other service providers.

4. Consideration on Marking Bits

The marking bits selection is encapsulation-related; different bits for marking should be allocated by different encapsulations. This document does not define any marking bits. The marking bits selection for specific encapsulations will be defined in the relevant documents. In general, at least one marking bit is required to support loss and delay measurement. Specifically, if the second delay measurement solution is used (see Section 3), then at least two marking bits are needed; one bit for packet loss measurement, the other for packet delay measurement.

In theory, so long as there are unused bits that could be allocated for marking purpose, the marking-based measurement mechanism can be applied to any encapsulation. It is relatively easier for new encapsulations to allocate marking bits. An example of such a case is Bit Indexed Explicit Replication (BIER). Two marking bits for passive performance measurement has been allocated in the BIER encapsulation [I-D.ietf-bier-mppls-encapsulation] (Section 3.). However, for sophisticated encapsulations, it is harder or even impossible to allocate bits for marking purpose. The IPv4 encapsulation is one of the examples. The IPv6 encapsulation is in a similar situation, but for IPv6, an alternative solution is to leverage the IPv6 extension header for marking.

Since marking will directly change some bits (of the header) of the real traffic packets, the marking operations MUST NOT affect the forwarding and processing of packets. Specifically, the marking bits MUST NOT be used for ECMP hashing. In addition, to increase the accuracy of measurement, hardware-based implementation is desired. Thus, the location of the marking bits SHOULD be easy for hardware implementation. For example, the marking bits would be best located at fixed positions in a packet header.

5. Reference Model and Functional Components

5.1. Reference Model

The outline of the measurement system of large-scale measurement platforms (LMAP) is introduced in [I-D.ietf-lmap-framework]. It describes the main functional components of the LMAP measurement system, and the interactions between the components. The Measurement Agent (MA) of IPFPM could be considered equivalent to the MA of LMAP. The Measurement Control Point (MCP) of IPFPM could be considered as the combined function of Controller and Collector. The IP Flow

Information eXport (IPFIX) [RFC7011] protocol is used for collecting the performance measurement data on the MAs and reporting to the MCP. The details are specified in the companion document [I-D.chen-ippm-ipfpm-report]. The control between MCP and MAs are left for future study. Figure 2 presents the reference model of IPFPM.

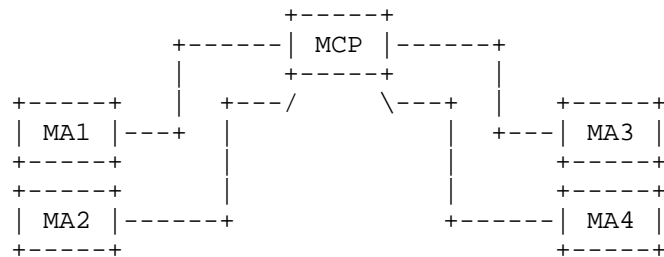


Figure 2: IPFPM Reference Model

5.2. Measurement Control Point

The Measurement Control Point (MCP) is responsible for collecting the measurement data from the Measurement Agents (MAs) and calculating the performance metrics according to the collected measurement data. For packet loss, based on each block of markers, the difference between the total counts received from all upstream MAs and the total counts received from all downstream MAs are the lost packet numbers. The MCP must make sure that the counts from the upstream MAs and downstream MAs are related to the same marking/packets block. For packet delay (e.g., one way delay), the difference between the timestamps from the downstream MA and upstream MA is the packet delay. Similarly to packet loss, the MCP must make sure the two timestamps are based on the same marked packet. This document introduces a Period Number (PN) based synchronization mechanism which is discussed in details in Section 6.

5.3. Measurement Agent

The Measurement Agent (MA) executes the measurement actions (e.g., marks the packets, counts the packets, records the timestamps, etc.), and reports the data to the Measurement Control Point (MCP). Each MA maintains two timers, one (C-timer, used at upstream MA) is for marking change, the other (R-timer, used at downstream MA) is for reading the packet counts and timestamps. The two timers have the same time interval but are started at different times. An MA can be either an upstream or a downstream MA: the role is specific to an IP flow to be measured. For a specific IP flow, the upstream MA will change the marking and read the packet counts and timestamps when the C-timer expires, the downstream MA just reads the packet counts and

timestamps when the R-timer expires. The MA may delay the reading for a certain time period when the R-timer expires, in order to be tolerant to a certain degree of packet re-ordering. Section 7 describes this in details.

For each Measurement Task (corresponding to an IP flow) [I-D.ietf-lmap-framework], an MA maintains a pair of packet counters and a timestamp counter for each block of markers. As for the pair of packet counters, one is for counting packets and the other is for counting octets.

6. Period Number

When data is collected on the upstream MA and downstream MA, e.g., packet counts or timestamps, and periodically reported to the MCP, a certain synchronization mechanism is required to ensure that the collected data is correlated. Synchronization aspects are further discussed in Section 10. This document introduces the Period Number (PN) to help the MCP to determine whether any two or more packet counts (from distributed MAs) are related to the same block of markers, or any two timestamps are related to the same marked packet.

Period Numbers assure the data correlation by literally splitting the packets into different measurement periods. The PN is generated each time an MA reads the packet counts or timestamps, and is associated with each packet count and timestamp reported to the MCP. For example, when the MCP sees two PNs associated with two packet counts from an upstream and a downstream MA, it assumes that these two packet counts correspond to the same measurement period by the same PN, i.e., that these two packet counts are related to the same block of markers. The assumption is that the upstream and downstream MAs are time synchronized. This requires the upstream and downstream MAs to have a certain time synchronization capability (e.g., the Network Time Protocol (NTP) [RFC5905], or the IEEE 1588 Precision Time Protocol (PTP) [IEEE1588]), as further discussed in Section 10. The PN is calculated as the modulo of the local time (when the counts or timestamps are read) and the interval of the marking time period.

7. Re-ordering Tolerance

In order to allow for a certain degree of packet re-ordering, the R-timer on downstream MAs should be started Δt (Dt) later than the C-timer is started. Dt is a defined period of time and should satisfy the following conditions:

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

Where

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

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

Thus, the R-timer should be started at " $t + Dt$ " (where t is the time at which C-timer is started).

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

8. Packet Loss Measurement

To simplify the process description, the flows discussed in this document are all unidirectional. A bidirectional flow can be seen as two unidirectional flows. For a specific flow, there will be an upstream MA and a downstream MA, and for each of these MAs there will be corresponding packet counts/timestamp.

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

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

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

U-CountO[n][m]: U-CountO is a two-dimensional array that stores the number of octets transmitted by each upstream MA in each marking time period. Specifically, parameter "n" is the "period number" of measured blocks of markers while parameter "m" refers to the m-th MA of the upstream MAs.

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

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

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

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

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

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

Where the m and m' are the number of upstream MAs and downstream MAs of the measured flow, respectively.

9. Packet Delay Measurement

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

U-Time[n]: U-Time is a one-dimension array that stores the time when marked packets are sent; in case the "average delay" method is being used, U-Time stores the average of the time when the packets of the same block are sent; parameter "n" is the "period number" of marked packets.

D-Time[n]: D-Time is a one-dimension array that stores the time when marked packets are received; in case the "average delay" method is being used, D-Time stores the average of the time when the packets of the same block are received; parameter "n" is the "period number" of marked packets. This is only for P2P model.

D-Time[n][m]: D-Time a two-dimension array that stores the time when the marked packet is received by downstream MAs at each marking time period; in case the "average delay" method is being used, D-Time stores the average of the times when the packets of the same block are received by downstream MAs at each marking time period. Here,

parameter "n" is the "period number" of marked packets while parameter "m" refers to the m-th MA of the downstream MAs. This is for P2MP model which is left for future study.

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

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

...

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

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

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

...

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

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

10. Synchronization Aspects

As noted in the previous sections, there are two mechanisms in IPFPM that require MAs to have synchronized clocks: (i) the period number (Section 6), and (ii) delay measurement.

This section elaborates on the level of synchronization that is required for each of the two mechanisms. Interestingly, IPFPM can be implemented even with very coarse-grained synchronization.

10.1. Synchronization for the Period Number

Period numbers are used to uniquely identify blocks, allowing the MCP to match the measurements of each block from multiple MAs.

The period number of each measurement is computed by the modulo of the local time. Therefore, if the length of the measurement period is L time units, then all MAs must be synchronized to the same clock reference with an accuracy of $\pm L/2$ time units. This level of accuracy guarantees that all MAs 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.

The synchronization requirement for maintaining the period number can be satisfied even with a relatively inaccurate synchronization method.

10.2. Synchronization for Delay Measurement

As discussed in Section 9, the delay between two MAs is computed by $D\text{-Time}[1] - U\text{-Time}[1]$, requiring the two MAs to be synchronized.

Notably, two-way delay measurement does not require the two MAs to be time synchronized. Therefore, a system that uses only two-way delay measurement does not require synchronization between MAs.

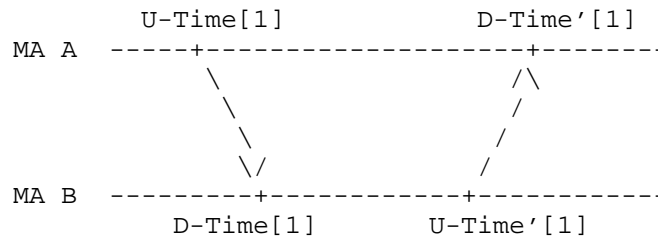


Figure 3: Two-way Delay Measurement

As shown in Section 9, the two way delay between two MAs is given by (see Figure 3):

$$(D\text{-Time}[1] - U\text{-Time}[1]) + (D\text{-Time}'[1] - U\text{-Time}'[1])$$

Therefore, the two-way delay is equal to:

$$(D\text{-Time}'[1] - U\text{-Time}[1]) - (U\text{-Time}'[1] - D\text{-Time}'[1])$$

The latter implies that the two-way delay is comprised of two time differences, $(D-Time'[1] - U-Time[1])$, and $(U-Time'[1] - D-Time'[1])$. Thus, the value of the clocks of MA A and MA B does not affect the computation, and synchronization is not required.

11. IANA Considerations

This document makes no request to IANA.

12. Security Considerations

This document specifies a passive mechanism for measuring packet loss and delay within a Service Provider's network where the IP packets are marked using unused bits in IP head field, thus avoiding the need to insert additional OAM packets during the measurement. Obviously, such a mechanism does not directly affect other applications running on the Internet but may potentially affect the measurement itself.

First, the measurement itself may be affected by routers (or other network devices) along the path of IP packets intentionally altering the value of marking bits of packets. As mentioned above, the mechanism specified in this document is just in the context of one Service Provider's network, and thus the routers (or other network devices) are locally administered and this type of attack can be avoided.

Second, 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 color bits are the only information that is exchanged between the MAs. Therefore, passive eavesdropping to data plane traffic does not allow attackers to gain information about the network performance. We note that the information exported from the MAs to the MCP can be subject to eavesdropping, and thus it should be encrypted.

Finally, 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].

13. Acknowledgements

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

14. Contributing Authors

Hongming Liu
Huawei Technologies

Email: liuhongming@huawei.com

Yuanbin Yin
Huawei Technologies

Email: yinyuanbin@huawei.com

Rajiv Papneja
Huawei Technologies

Email: Rajiv.Papneja@huawei.com

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

Email: shailesh.abhyankar@vodafone.com

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

Email: dengguangqing@cnnic.cn

Yongliang Huang
China Unicom

Email: huangyl@dipmt.com

15. References

15.1. 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>>.

15.2. Informative References

- [I-D.chen-ippm-ipfpm-report]
Chen, M., Zheng, L., Liu, H., Yin, Y., Papneja, R., Abhyankar, S., Deng, G., and Y. Huang, "IP Flow Performance Measurement Report", draft-chen-ippm-ipfpm-report-00 (work in progress), July 2014.
- [I-D.deng-ippm-passive-wireless-usecase]
Lingli, D., Zheng, L., and G. Mirsky, "Use-cases for Passive Measurement in Wireless Networks", draft-deng-ippm-passive-wireless-usecase-01 (work in progress), January 2015.
- [I-D.ietf-bier-mpls-encapsulation]
Wijnands, I., Rosen, E., Dolganow, A., Tantsura, J., and S. Aldrin, "Encapsulation for Bit Index Explicit Replication in MPLS Networks", draft-ietf-bier-mpls-encapsulation-03 (work in progress), February 2016.
- [I-D.ietf-lmap-framework]
Eardley, P., Morton, A., Bagnulo, M., Burbidge, T., Aitken, P., and A. Akhter, "A framework for Large-Scale Measurement of Broadband Performance (LMAP)", draft-ietf-lmap-framework-14 (work in progress), April 2015.
- [I-D.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.
- [IEEE1588]
IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<http://www.rfc-editor.org/info/rfc2474>>.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, DOI 10.17487/RFC2679, September 1999, <<http://www.rfc-editor.org/info/rfc2679>>.
- [RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, DOI 10.17487/RFC3260, April 2002, <<http://www.rfc-editor.org/info/rfc3260>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<http://www.rfc-editor.org/info/rfc4656>>.
- [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>>.
- [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>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<http://www.rfc-editor.org/info/rfc6374>>.
- [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, <<http://www.rfc-editor.org/info/rfc7011>>.
- [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

Mach(Guoyi) Chen (editor)
Huawei Technologies

Email: mach.chen@huawei.com

Lianshu Zheng (editor)
Huawei Technologies

Email: vero.zheng@huawei.com

Greg Mirsky (editor)
Ericsson
USA

Email: gregory.mirsky@ericsson.com

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

Email: giuseppe.fioccola@telecomitalia.it

Tal Mizrahi (editor)
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 27, 2013

J. Hedin
G. Mirsky
Ericsson
June 25, 2013

Type-P Descriptor Monitoring in Two-Way Active Measurement Protocol
(TWAMP)
draft-hedin-ippm-type-p-monitor-01

Abstract

This document specifies how optional monitoring of Type-P Descriptor can be negotiated and performed by TWAMP [RFC5357] Control and Test protocols.

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 27, 2013.

Copyright Notice

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

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

Table of Contents

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

1. Introduction

Re-marking of Type-P Descriptor, i.e. change in value, might be demonstration of intentional or erroneous behavior. Monitoring of Type-P Descriptor can provide valuable information for network operators. One-Way Active Measurement Protocol [RFC4656] and Two-Way Active Measurement Protocol [RFC5357] define negotiation of TypeP Descriptor value that must be used by Session-Sender and Session-Reflector. But there's not means for Session-Sender to know whether Type-P Descriptor was received by Session-Reflector unchanged. Opional monitoring of Type-P Descriptor between Session-Sender and Session-Reflector requires extensions to TWAMP [RFC5357] that are described in this document.

1.1. Conventions used in this document

1.1.1. Terminology

DSCP: Differentiated Service Codepoint

IPPM: IP Performance Measurement

TWAMP: Two-Way Active Measuremnt Protocol

OWAMP: One-Way Active Measurement Protocol

1.1.2. Requirements Language

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

2. TWAMP Extensions

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

2.1. Setting Up Connection to Monitor Type-P Descriptor

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

2.2. TWAMP-Test Extension

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

2.2.1. Session-Reflector Packet Format for Type-P Descriptor Monitoring

When Session-Reflector supports Type-P Descriptor Monitoring in MUST construct Sender Type-P Descriptor for each test packet it sends to Session-Sender according to the following procedure:

- first two bits MUST be the same as two first bits of Type-P Descriptor field Request-Session control packet;
- remaining bits MUST be copied from received Session-Sender test packet according to two first bits:

Section 3.5 in [RFC5357] states that Type-P Descriptor capability supported in TWAMP is to set Differentiated Services Codepoint (DSCP) value, as defined in [RFC2474]. Thus first two bits MUST be set to 00. Then DSCP value copied into subsequent six bits. For a Session-Sender, upon receiving reflected TWAMP-Test packet, If the first two bits are not 00, then subsequent value should be ignored.

For unauthenticated mode:

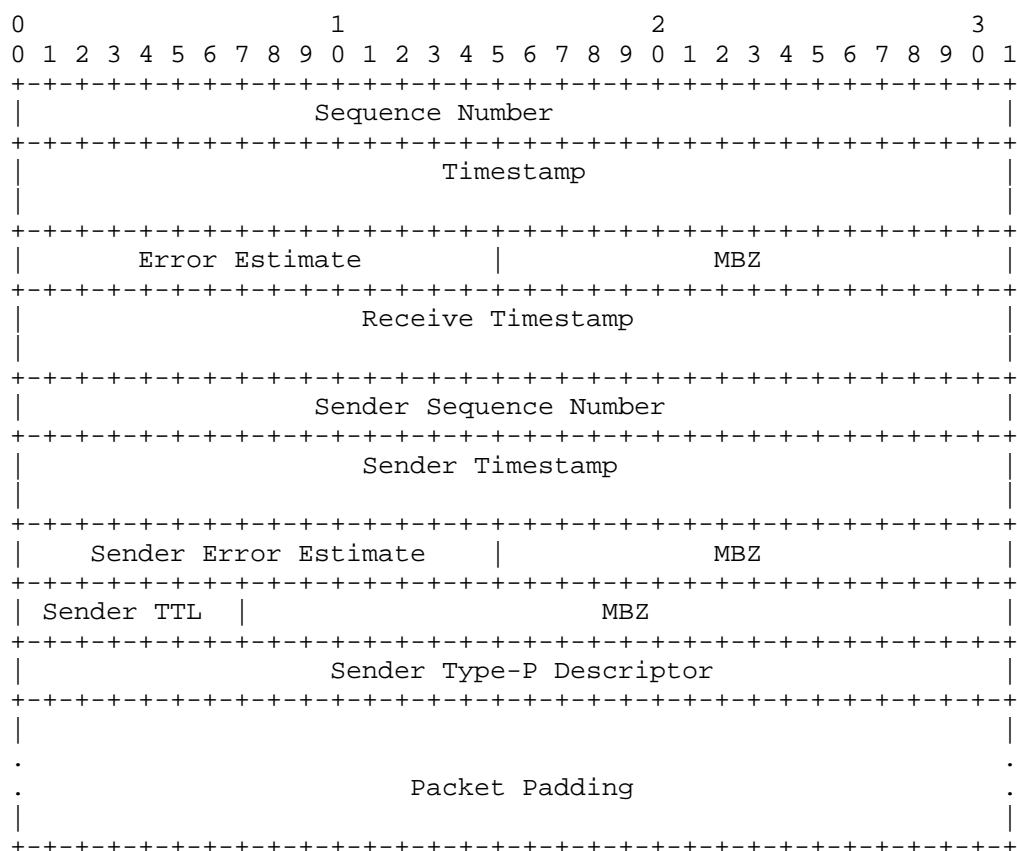
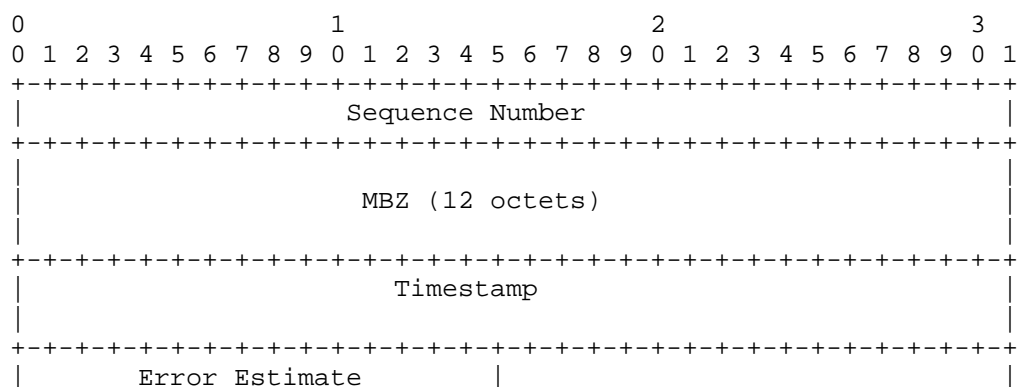


Figure 1: Session-Reflector test packet format with Type-P Descriptor monitoring in unauthenticated mode

For authenticated and encrypted modes:



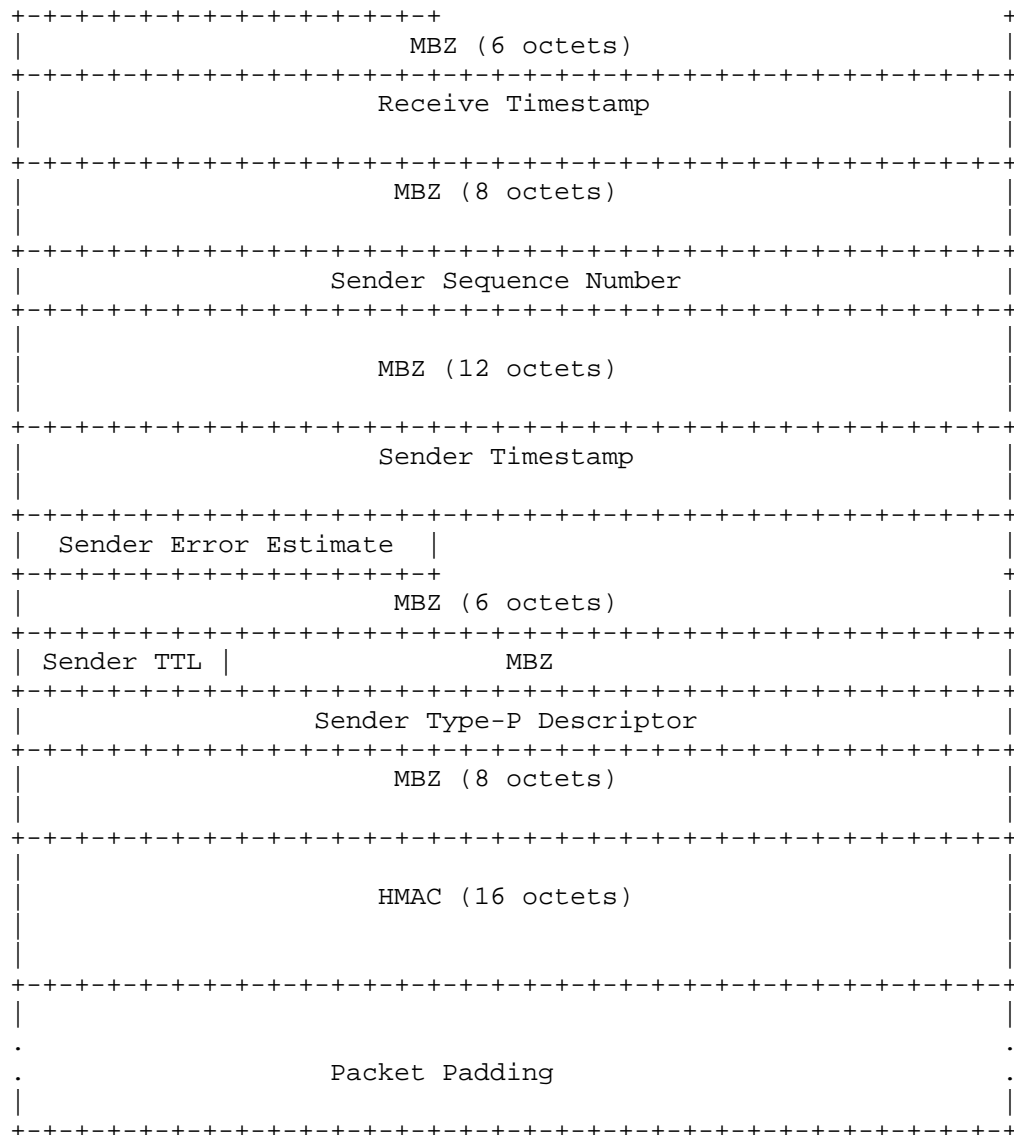


Figure 2: Session-Reflector test packet format with Type-P Descriptor monitoring in authenticated or encrypted modes

2.2.2. Type-P Descriptor Monitoring with RFC 6038 extensions

[RFC6038] defined two extensions to TWAMP. First, to ensure that Session-Sender and Session-Reflector exchange TWAMP-Test packets of equal size. Second, to specify number of octets to be reflected by

Session-Reflector. If Type-P Descriptor monitoring and Symmetrical Size and/or Reflects Octets modes being negotiated between Server and Control-Client in Unauthenticated mode, then because Sender Type-P Descriptor increases size of unauthenticated Session-Reflector packet by 4 octets the Padding Length value SHOULD be ≥ 31 octets to allow for the truncation process that TWAMP recommends in Section 4.2. 1 of [RFC5357].

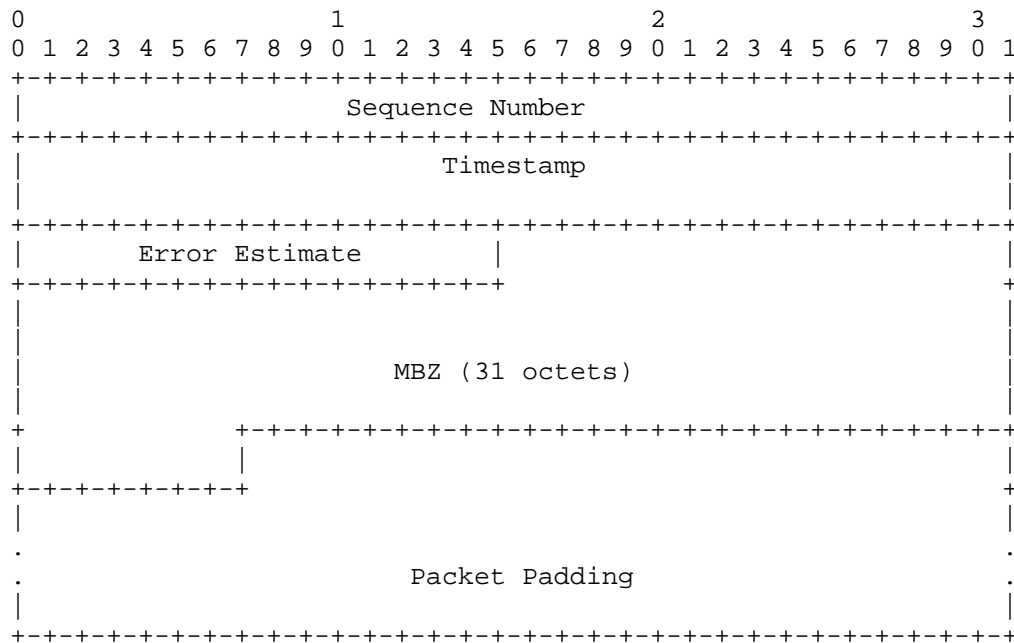


Figure 3: Session-Sender test packet format with Type-P Descriptor monitoring and Symmetrical Test Packet in unauthenticated mode

3. IANA Considerations

The TWAMP-Modes registry defined in [RFC5618].

IANA is requested to reserve a new Type-P Descriptor Monitoring Capability as follows:

Value	Description	Semantics	Reference
X (proposed 128)	Type-P Descriptor Monitoring Capability	bit position Y (proposed 7)	This document

Table 1: New Type-P Descriptor Monitoring Capability

4. Security Considerations

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

5. Acknowledgements

TBD

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618,

August 2009.

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

6.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Jonas Hedin
Ericsson

Email: jonas.hedin@ericsson.com

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 11, 2014

J. Fabini
Vienna University of Technology
A. Morton
AT&T Labs
July 10, 2013

Advanced Stream and Sampling Framework for IPPM
draft-ietf-ippm-2330-update-00

Abstract

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

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on January 11, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Definition: Reactive Network Behavior	3
2. Scope	4
3. New Stream Parameters	4
3.1. Test Packet Type-P	5
3.1.1. Test Packet Length	6
3.1.2. Test Packet Payload Content Optimization	6
3.2. Packet History	6
3.3. Access Technology Change	7
3.4. Time-Slotted Randomness Cancellation	7
4. Conclusions	8
5. Security Considerations	9
6. IANA Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Authors' Addresses	11

1. Introduction

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

The IPPM framework [RFC2330] generally relies on several assumptions, one of which is not explicitly stated but assumed: the network behaves (halfway) deterministic and without state/history-less (with some exceptions, firewalls are mentioned). However, this does not hold true for many modern network technologies, such as reactive networks (those with demand-driven resource allocation) and links with time-slotted operation. Per-flow state can be observed on test packet streams, and such treatment will influence network characterization if it is not taken into account. Flow history will also affect the performance of applications and be perceived by their users.

Moreover, Sections 4 and 6.2 of [RFC2330] explicitly recommend repeatable measurement metrics and methodologies. Measurements in today's access networks illustrate that methodological guidelines of [RFC2330] must be extended to capture the reactive nature of these networks. Although the proposed extensions can support methodologies to fulfill the continuity requirement stated in section 6.2 of [RFC2330], there is no guarantee. Practical measurements confirm that some link types exhibit distinct responses to repeated measurements with identical stimulus, i.e., identical traffic patterns. If feasible, appropriate fine-tuning of measurement traffic patterns can improve measurement continuity and repeatability for these link types as shown in [IBD].

1.1. Definition: Reactive Network Behavior

A network or network path is defined to be reactive when at least one of the links or hosts in it exhibits reactive behavior. Reactive behavior is present when link-or host-internal sensing (measurement) of packet arrival for the flow of interest indicates that traffic is absent or present, or that traffic during a measurement interval is above or below a threshold, and the results of one or more successive measurements cause one or more network components to process future packets using a different mode of operation than for other measurement outcomes.

Reactive network behavior must be observable by the test packet stream as a repeatable phenomenon where packet transfer performance characteristics *change* according to prior node- or link-internal observations of the packet flow of interest. Therefore, reactive

network behavior is deterministic with respect to the flow of interest. Other flows or traffic load conditions may result in additional performance-affecting reactions, but these are external to the characteristics of the flow of interest.

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

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

2. Scope

The scope of this memo is to describe useful stream parameters in addition to the information in Section 11.1 of [RFC2330] and described in [RFC3432] for periodic streams. The purpose is to foster repeatable measurement results in modern networks by highlighting the key aspects of test streams and packets and make them part of the IPPM performance metric framework.

3. New Stream Parameters

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

1. Network treatment depends on the fullest extent on the "packet of Type-P" definition in [RFC2330], and has for some time.
 - * State is often maintained on the per-flow basis at various points in the network, where "flows" are determined by IP and other layers. Significant treatment differences occur with the simplest of Type-P parameters: packet length.
 - * Payload content optimization (compression or format conversion) in intermediate segments. This breaks the convention of payload correspondence when correlating measurements made at different points in a path.

2. Packet history (instantaneous or recent test rate or inactivity, also for non-test traffic) profoundly influences measured performance, in addition to all the Type-P parameters described in [RFC2330].
3. Access technology may change during testing. A range of transfer capacities and access methods may be encountered during a test session. When different interfaces are used, the host seeking access will be aware of the technology change which differentiates this form of path change from other changes in network state. Section 14 of [RFC2330] treats the possibility that a host may have more than one attachment to the network, and also that assessment of the measurement path (route) is valid for some length of time (in Section 5 and Section 7 of [RFC2330]). Here we combine these two considerations under the assumption that changes may be more frequent and possibly have greater consequences on performance metrics.
4. Paths including links or nodes with time-slotted service opportunities represent several challenges to measurement (when service time period is appreciable):
 - * Random/unbiased sampling is not possible beyond one such link in the path.
 - * The above encourages a segmented approach to end to end measurement, as described in [RFC6049] for Network Characterization (as defined in [RFC6703]) to understand the full range of delay and delay variation on the path. Alternatively, if application performance estimation is the goal (also defined in [RFC6703]), then a stream with un-biased or known-bias properties [RFC3432] may be sufficient.
 - * Multi-modal delay variation makes central statistics unimportant, others must be used instead.

Each of these topics is treated in detail below.

3.1. Test Packet Type-P

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

3.1.1. Test Packet Length

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

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

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

3.1.2. Test Packet Payload Content Optimization

The aim for efficient network resource use has resulted in a series of "smart" networks to deploy server-only or client-server lossless or lossy payload compression techniques on some links or paths. These optimizers attempt to compress high-volume traffic in order to reduce network load. Files are analyzed by application-layer parsers and parts (like comments) might be dropped. Although typically acting on HTTP or JPEG files, compression might affect measurement packets, too. In particular measurement packets are qualified for efficient compression when they use standard plain-text payload.

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

3.2. Packet History

Recent packet history and instantaneous data rate influence measurement results for reactive links supporting on-demand capacity allocation. Measurement uncertainty may be reduced by knowledge of

measurement packet history and total host load. Additionally, small changes in history, e.g., because of lost packets along the path, can be the cause of large performance variations.

For instance delay in reactive 3G networks like High Speed Packet Access (HSPA) depends to a large extent on the test traffic data rate. The reactive resource allocation strategy in these networks affects the uplink direction in particular. Small changes in data rate can be the reason of more than 200% increase in delay, depending on the specific packet size.

3.3. Access Technology Change

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

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

3.4. Time-Slotted Randomness Cancellation

Time-Slotted operation of network entities - interfaces, routers or links - in a network path is a particular challenge for measurements, especially if the time slot period is substantial. The central observation as an extension to Poisson stream sampling in [RFC2330] is that the first such time-slotted component cancels unbiased measurement stream sampling. In the worst case, time-slotted operation converts an unbiased, random measurement packet stream into a periodic packet stream. Being heavily biased, these packets may interact with periodic network behavior of subsequent time-slotted network entities[TSRC].

Time-slotted randomness cancellation (TSRC) sources can be found in virtually any system, network component or path, their impact on measurements being a matter of the order of magnitude when compared to the metric under observation. Examples of TSRC sources include but are not limited to system clock resolution, operating system ticks, time-slotted component or network operation, etc. The amount of measurement bias is determined by the particular measurement stream, relative offset between allocated time-slots in subsequent

network entities, delay variation in these networks, and other sources of variation. Measurement results might change over time, depending on how accurately the sending host, receiving host, and time-slotted components in the measurement path are synchronized to each other and to global time. If network segments maintain flow state, flow parameter change or flow re-allocations can cause substantial variation in measurement results.

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

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

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

4. Conclusions

Safeguarding continuity and repeatability as key properties of measurement methodologies is highly challenging and sometimes impossible in reactive networks. Measurements in networks with demand-driven allocation strategies must use a prototypical application packet stream to infer a specific application's performance. Measurement repetition with unbiased network and flow states (e.g., by rebooting measurement hosts) can help to avoid interference with periodic network behavior, randomness being a mandatory feature for avoiding correlation with network timing. Inferring the network performance between one measurement session or packet stream and other streams with alternate characteristics is generally discouraged with reactive networks because of the huge set

of global parameters which have influence instantaneous network performance.

5. Security Considerations

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

6. IANA Considerations

This memo makes no requests of IANA.

7. Acknowledgements

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

8. References

8.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol

(OWAMP)", RFC 4656, September 2006.

- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

8.2. Informative References

- [I-D.ietf-bmwg-imix-genome]
Morton, A., "IMIX Genome: Specification of variable packet sizes for additional testing", draft-ietf-bmwg-imix-genome-05 (work in progress), June 2013.
- [IBD] Fabini, J., "The Illusion of Being Deterministic - Application-Level Considerations on Delay in 3G HSPA Networks", Lecture Notes in Computer Science, Springer, Volume 5550, 2009, pp 301-312 , May 2009.
- [IRR] Fabini, J., "The Importance of Being Really Random: Methodological Aspects of IP-Layer 2G and 3G Network Delay Assessment", ICC'09 Proceedings of the 2009 IEEE International Conference on Communications, doi: 10.1109/ICC.2009.5199514, June 2009.
- [TSRC] Fabini, J., "Delay Measurement Methodology Revisited: Time-slotted Randomness Cancellation", IEEE Transactions on Instrumentation and Measurement doi:10.1109/TIM.2013.2263914, July 2013.

Authors' Addresses

Joachim Fabini
Vienna University of Technology
Favoritenstrasse 9/E389
Vienna, 1040
Austria

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

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

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

IPPM WG
Internet-Draft
Intended status: Standards Track
Expires: January 06, 2014

K. Pentikousis, Ed.
Y. Cui
E. Zhang
Huawei Technologies
July 05, 2013

Network Performance Measurement for IPsec
draft-ietf-ippm-ipsec-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 06, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Motivation	4
3.1. O/TWAMP-Control Security	5
3.2. O/TWAMP-Test Security	6
3.3. O/TWAMP Security Root	6
3.4. O/TWAMP and IPsec	7
4. O/TWAMP for IPsec Networks	8
4.1. Shared Key Derivation	8
4.2. Optimizations	10
4.2.1. Alternative 1	11
4.2.2. Alternative 2	13
5. Security Considerations	14
6. IANA Considerations	14
7. Acknowledgments	14
8. References	14
8.1. Normative References	14
8.2. Informative References	15
Authors' Addresses	15

1. Introduction

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

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

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

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

2. Terminology

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

3. Motivation

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

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

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

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

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

OWAMP and TWAMP actually consist of two inter-related protocols: O/TWAMP-Control and O/TWAMP-Test. With respect to TWAMP, since "TWAMP

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

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

3.1. O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on

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

Three modes of operation are supported: unauthenticated, authenticated, and encrypted. The authenticated and encrypted modes require that endpoints possess a shared secret, typically a passphrase. The secret key is derived from the passphrase using a password-based key derivation function PBKDF2 (PKCS#5) [RFC2898].

In the unauthenticated mode, the security parameters are left unused. In the authenticated and encrypted modes, security parameters are negotiated during the control connection establishment. In short, the client opens a TCP connection to the server in order to be able to send OWAMP-Control commands. The server responds with a server greeting, which contains the Challenge, Mode, Salt and Count. If the client-requested mode is available, the client responds with a Set-Up-Response message, wherein the KeyID, Token and Client IV are included. The Token is the concatenation of a 16-octet challenge, a 16-octet AES Session-key used for encryption, and a 32-octet HMAC-SHA1 Session-key used for authentication. The Token is encrypted using AES-CBC.

Encryption uses a key derived from the shared secret associated with KeyID. In the authenticated and encrypted modes, all further communication is encrypted using the AES Session-key and authenticated with the HMAC Session-key. The client encrypts everything it transmits through the just-established O/TWAMP-Control connection using stream encryption with Client-IV as the IV. Correspondingly, the server encrypts its side of the connection using Server-IV as the IV. The IVs themselves are transmitted in cleartext. Encryption starts with the block immediately following that containing the IV.

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

3.2. O/TWAMP-Test Security

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

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

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

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

3.3. O/TWAMP Security Root

As discussed above, the AES Session-key and HMAC Session-key used in the O/TWAMP-Test protocol are derived from the AES Session-key and HMAC Session-key which are used in O/TWAMP-Control protocol. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the client, and encrypted with the shared secret associated with KeyID. Therefore, the security root is the shared secret key. Thus, key provision and management may become overly complicated. Comparatively, a certificate-based approach using IKEv2/IPsec can automatically manage the security root and solve this problem, as we explain in Section 4.

3.4. O/TWAMP and IPsec

According to RFC 4656 the "deployment paths of IPsec and OWAMP could be separate if OWAMP does not depend on IPsec." However, the problem that arises in practice is that the security mechanism of O/TWAMP and IPsec cannot coexist at the same time without adding overhead or increasing complexity.

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

When sender and receiver implement O/TWAMP over IPsec, they need to agree on a shared secret key during the IPsec tunnel establishment. Subsequently, all IP packets sent by the sender are protected. If the AH protocol is used, IP packets are transmitted in plaintext. The authentication part covers the entire packet. So all test information, such as UDP port number, and the test results will be visible to any attacker, which can intercept these test packets, and introduce errors or forge packets that may be injected during the transmission. In order to avoid this attack, the receiver must validate the integrity of these packets with the negotiated secret key. If ESP is used, IP packets are encrypted, and hence only the receiver can use the IPsec secret key to decrypt the IP packet, and obtain the test data in order to assess the IP network performance based on the measurements. Both the sender and receiver must support IPsec to generate the security secret key of IPsec.

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

Of course, backward compatibility should be considered as well. That is, the intrinsic security method based on shared key as specified in the O/TWAMP standards can also still be suitable for other network settings. There should be no impact on the current security mechanisms defined in O/TWAMP for other use cases. This document describes possible solutions to this problem which take advantage of the secret key derived by IPsec, in order to provision the key needed for active network measurements based on RFC 4656 and RFC 5357.

4. O/TWAMP for IPsec Networks

This section presents a method of binding O/TWAMP and IKEv2 for network measurements between a sender and a receiver which both support IPsec. In short, the shared key used for securing O/TWAMP traffic is derived using IKEv2 [RFC5996].

4.1. Shared Key Derivation

If the AH protocol is used, the IP packets are transmitted in plaintext, but all O/TWAMP traffic is integrity-protected by IPsec. Therefore, even if the peers choose to opt for the unauthenticated mode, IPsec integrity protection is extended to O/TWAMP.

In the authenticated and encrypted modes, the shared secret can be derived from the IKEv2 Security Association (SA), or IPsec SA. If the shared secret key is derived from the IKEv2 SA, SKEYSEED must be generated firstly.

SKEYSEED and its derivatives are computed as per [RFC5996], where prf is a pseudorandom function:

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

Ni and Nr are nonces negotiated during the initial exchange. g^{ir} is the shared secret from the ephemeral Diffie-Hellman exchange and is represented as a string of octets. Note that this SKEYSEED can be used as the O/TWAMP shared secret key directly.

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

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

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

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

$$\text{KEYMAT} = \text{prf+}(\text{SK_d}, \text{Ni} \parallel \text{Nr})$$

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

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

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

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

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

The SPI and protocol type are included in the Server Greeting of the O/TWAMP-Control protocol (Figure 1). After the client receives the greeting, it MUST close the connection if it receives a greeting with an erroneous SPI and protocol value (Figure 2). Otherwise, the client SHOULD respond with the following Set-Up-Response message and generates the shared secret key.

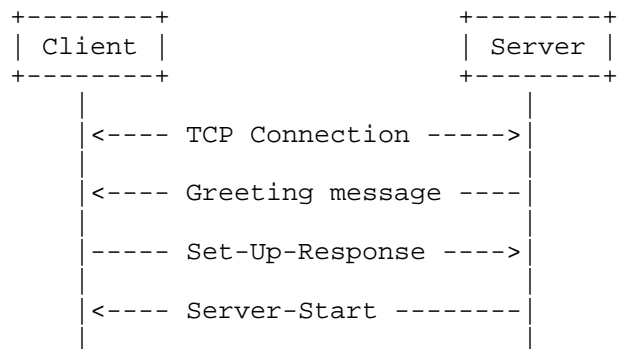
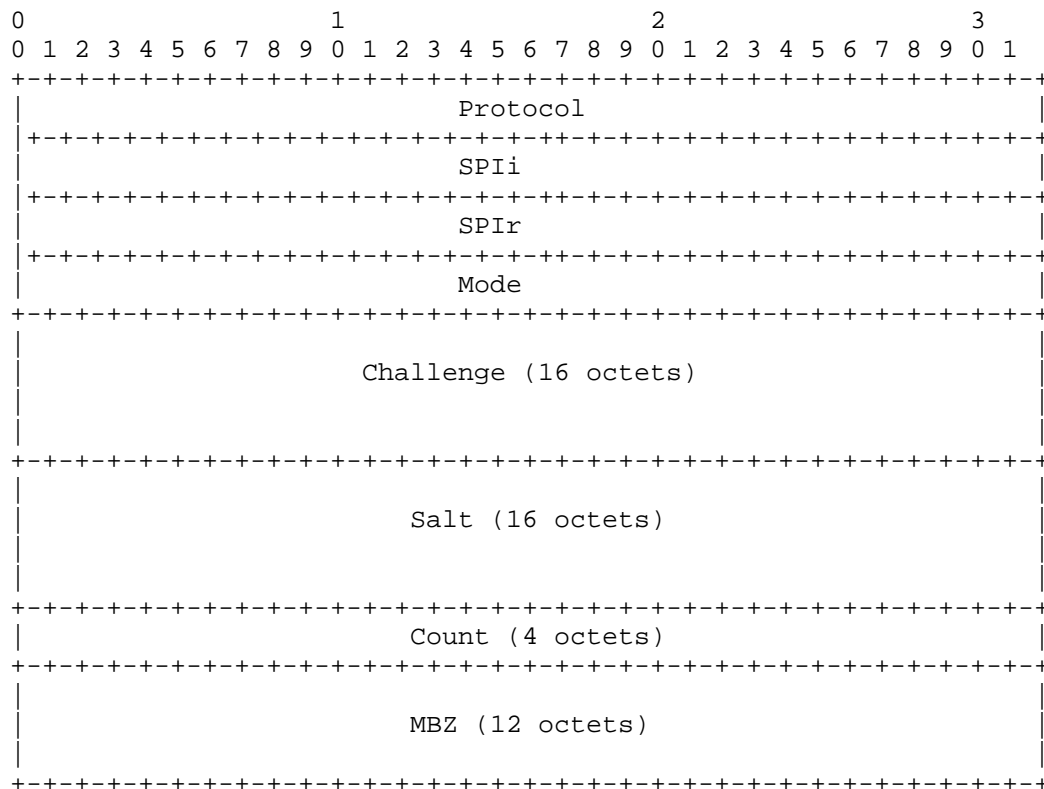


Figure 1: Initiation of O/TWAMP-Control

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



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

4.2. Optimizations

The previous subsection described a method for deriving the shared key for O/TWAMP by capitalizing on IPsec. We note, however, that the O/TWAMP protocol uses distinct encryption and integrity keys for O/TWAMP-Control and O/TWAMP-Test. Consequently, four keys are generated to protect O/TWAMP-Control and O/TWAMP-Test messages.

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

EDITOR'S NOTE:

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

4.2.1. Alternative 1

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

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

If the root key of O/TWAMP is derived from the IPsec SA, the shared secret key can be equal to KEYMAT. KEYMAT and its derivatives are computed as per usual [RFC5996]. Then, the session keys for encryption and authentication can be derived from the root key of O/TWAMP, wherein:

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

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

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

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

In this optimization, the O/TWAMP-Control message exchange flow remains as per Figure 1. However, the optimized Server Greeting (Figure 3) can do without the Salt and Count parameters (cf. Figure 2) since the root key of O/TWAMP is derived from IKE SA or IPsec SA. O/TWAMP security can rely on IPsec and the SPI can uniquely identify the IPsec SA from which the root key was derived from.

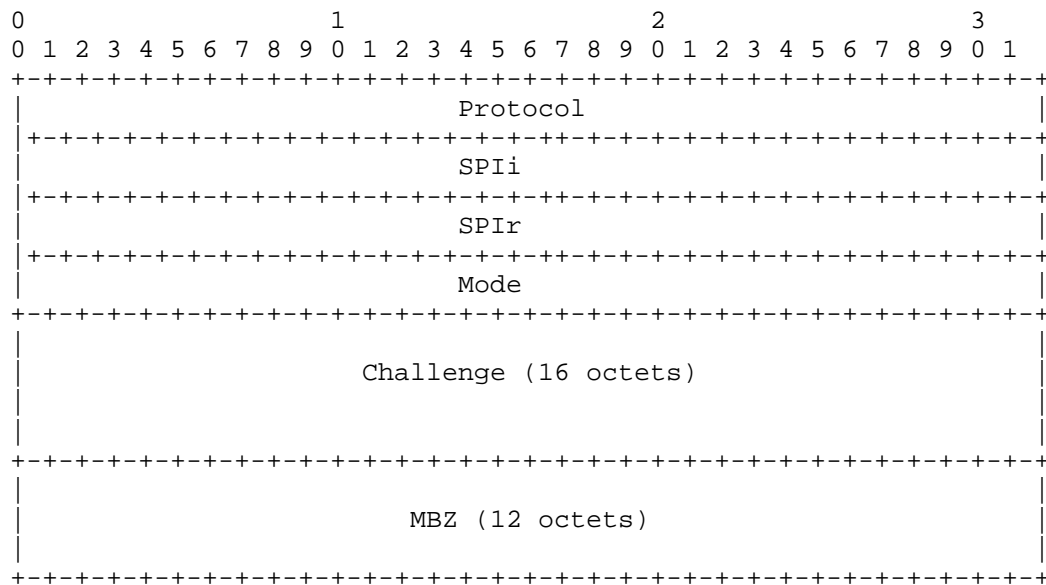


Figure 3: Optimized Server Greeting format

The format of the Set-Up-Response is illustrated in Figure 4. The Token carried in the Set-Up-Response is calculated as follows:

```
Token = Enc_root-key( Challenge )
```

where Challenge is the value received earlier in the Server Greeting (Figure 3)

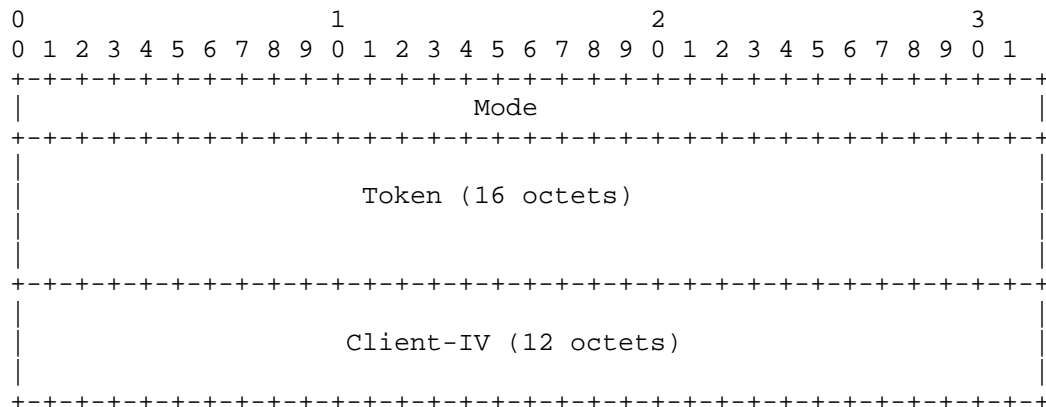
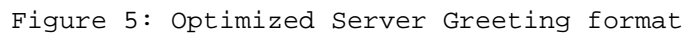


Figure 4: Set-Up-Response in Alternative 1

4.2.2. Alternative 2

Session key for enc = encryption key of the IPsec SA

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



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

5. Security Considerations

As the shared secret key is derived from IPsec, the key derivation algorithm strength and limitations are as per [RFC5996]. The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator employed. The strength of all keys and implementation vulnerabilities, particularly Denial of Service (DoS) attacks are as defined in [RFC5996].

EDITOR'S NOTE:

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

6. IANA Considerations

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

7. Acknowledgments

Emily Bi contributed to an earlier version of this document.

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

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

8.2. Informative References

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

Authors' Addresses

Kostas Pentikousis (editor)
Huawei Technologies
Carnotstrasse 4
10587 Berlin
Germany

Email: k.pentikousis@huawei.com

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

Email: cuiyang@huawei.com

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

Email: emma.zhanglijia@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 09, 2014

M. Bagnulo
UC3M
T. Burbridge
BT
S. Crawford
SamKnows
P. Eardley
BT
A. Morton
AT&T Labs
July 08, 2013

A Reference Path and Measurement Points for LMAP
draft-ietf-ippm-lmap-path-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 09, 2014.

Copyright Notice

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

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

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

Table of Contents

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

1. Introduction

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

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

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

The motivation for this memo is to provide an unambiguous framework to describe measurement coverage, or scope of the reference path. This is an essential part of the metadata to describe measurement results. Measurements conducted over different path scopes are not a valid basis for performance comparisons.

2. Purpose and Scope

The scope of this memo is to define a reference path for LMAP activities with sufficient level of detail to determine the location of different measurement points without ambiguity.

The bridge between the reference path and specific network technologies (with differing underlying architectures) is within the scope of this effort. Both wired and wireless technologies are in-scope.

The purpose is to create an efficient way to describe the location of the measurement point(s) used to conduct a particular measurement so that the measurement result will adequately described in this regard. This should serve many measurement uses, including diagnostic (where the same metric may be measured over many different path scopes) and comparative (where the same metric may be measured on different network infrastructures).

3. Terms and Definitions

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

3.1. Reference Path

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

3.2. Subscriber

An entity possessing one or more hosts participating in an Internet access service.

3.3. Dedicated Component (Links or Nodes)

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

3.4. Shared Component (Links or Nodes)

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

3.5. Resource Transition Point

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

4. Reference Path

This section defines a reference path for Internet Access.

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

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

GRA = Globally Routable Address, GW = Gateway

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

- o Subsc. (Subscriber) device - This is a host that normally originates and terminates communications conducted over the IP packet transfer service.

- o Private Net #x - This is a network of devices owned and operated by the Internet Access Service Subscriber. In some configurations, one or more private networks and the device that provides the Access Service Demarcation point are collapsed in a single device (and ownership may shift to the service provider), and this should be noted as part of the path description.
- o Access (Service) Demarcation point - this varies by technology but is usually defined as the Ethernet interface on a residential gateway or modem where the scope of access packet transfer service begins and ends. In the case of a WiFi Service, this would be an Air Interface within the intended service boundary (e.g., walls of the coffee shop). The Demarcation point may be within an integrated endpoint using an Air Interface (e.g., LTE UE). Ownership may not affect the demarcation point; a Subscriber may own all equipment on their premises, but it is likely that the service provider will certify such equipment for connection to their access network, or a third-party will certify standards compliance.
- o Intra IP Access - This is the first point in the access architecture beyond the Access Service Demarc. where a globally routable IP address is exposed and used for routing. In architectures that use tunneling, this point may be equivalent to the GRA GW. This point could also collapse to the device providing the Access Service Demarc., in principle. Only one Intra IP Access point is shown, but they can be identified in any access or transit network.
- o GRA GW - the point of interconnection between the access administrative domain and the rest of the Internet, where routing will depend on the GRAs in the IP header.
- o Transit GRA GW - Networks that intervene between the Subscriber's Access network and the Destination Host's network are designated "transit" and involve two GRA GW.

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

In the case that a private address space is used throughout an access architecture, then the Access Service Demarc. and the Intra IP Access points must use the same address space and be separated by the shared and dedicated access link infrastructure, such that a test between these points produces a useful assessment of access performance.

5. Measurement Points

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

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

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

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

...	Transit	--	GRA	--	Service	--	Private	--	Private	--	Destination
	GRA GW		GW		Demarc.		Net #n		Net #n+1		Host
	mpX90		mp890		mp800						mp900

GRA = Globally Routable Address, GW = Gateway

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

Notes:

- o Some use the terminology "on-net" and "off-net" when referring to Internet Service Provider (ISP) measurement coverage. With respect to the reference path, tests between mp100 and mp190 are "on-net".
- o Widely deployed broadband access measurements have used pass-through devices[SK] (at the subscriber's location) directly connected to the service demarcation point: this would be located at mp100.
- o The networking technology used at all measurement points must be indicated, especially the interface standard and configured speed.

- o If it can be shown that a link connecting to a measurement point has reliably deterministic or negligible performance, then the remote end of the connecting link is an equivalent point for some methods of measurement (To Be Specified Elsewhere). In any case, the presence of such a link must be reported.
- o Many access network architectures have a traffic aggregation point (e.g., CMTS or DSLAM) between mp100 and mp150. We designate this point mp120, but it won't currently fit in the figure.
- o A Carrier Grade NAT (CGN) deployed in the Subscriber's access network would be positioned between mp100 and mp190, and the egress side of the CGN will typically be designated mp150.
- o In the case that a private address space is used in an access architecture, then mp100 may need to use the same address space as its remote measurement point counterpart, so that a test between these points produces a useful assessment of network performance. Tests between mp000 and mp100 could use private address space, and when the egress side of a CGN is at mp150, then the private address side of the CGN could be designated mp149 for tests with mp100.
- o Measurement points at Transit GRA GWs are numbered mpX00 and mpX90, where X is the lowest positive integer not already used in the path.

6. Translation Between Ref. Path and Tech. X

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

We provide an example for 3G Cellular access below.

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

|_____UE_____||____RAN+Core_____|____GGSN____|

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

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

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

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

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

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

GRA = Globally Routable Address, GW = Gateway

Consider next the case where:

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

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

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

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

```
|--UE--|-----CPE/NAT-----|-----| -CGN- |-----|  
                                |---Access Network--|
```

GRA = Globally Routable Address, GW = Gateway

7. Security considerations

Specification of a Reference Path and identification of measurement points on the path represent agreements among interested parties, and they present no threat to the readers of this memo or to the Internet itself.

8. IANA Considerations

TBD

9. Acknowledgements

Thanks to Matt Mathis for review and comments.

10. References

10.1. Normative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, August 2012.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.

10.2. Informative References

- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, April 2011.
- [SK] Crawford, Sam., "Test Methodology White Paper", SamKnows Whitebox Briefing Note
<http://www.samknows.com/broadband/index.php>, July 2011.

Authors' Addresses

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

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

Trevor Burbridge
British Telecom
Adastral Park, Martlesham Heath
IPswitch
ENGLAND

Email: trevor.burbridge@bt.com

Sam Crawford
SamKnows

Email: sam@samknows.com

Phil Eardley
British Telecom
Adastral Park, Martlesham Heath
IPswitch
ENGLAND

Email: philip.eardley@bt.com

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

Email: acmorton@att.com

IP Performance Working Group
Internet-Draft
Intended status: Experimental
Expires: December 23, 2013

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

Model Based Bulk Performance Metrics
draft-ietf-ippm-model-based-metrics-00.txt

Abstract

We introduce a new class of model based metrics designed to determine if a long path can meet predefined end-to-end application performance targets. This is done by subpath at a time testing -- by applying a suite of single property tests to successive subpaths of a long path. In many cases these single property tests are based on existing IPPM metrics, with the addition of success and validity criteria. The subpath at a time tests are designed to facilitate IP providers eliminating all known conditions that might prevent the full end-to-end path from meeting the users target performance.

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

A natural consequence of this methodology is verifiable network measurement: measurements from any given vantage point are repeatable from other vantage points.

Formatted: Fri Jun 21 18:23:29 PDT 2013

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months

and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 23, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	5
1.1. TODO	6
2. Terminology	6
3. New requirements relative to RFC 2330	8
4. Background	9
4.1. TCP properties	11
5. Common Models and Parameters	12
5.1. Target End-to-end parameters	13
5.2. Common Model Calculations	13
5.3. Parameter Derating	14
6. Common testing procedures	15
6.1. Traffic generating techniques	15
6.1.1. Paced transmission	15
6.1.2. Constant window pseudo CBR	16
6.1.2.1. Scanned window pseudo CBR	16
6.1.3. Intermittent Testing	16
6.1.4. Intermittent Scatter Testing	17
6.2. Interpreting the Results	17
6.2.1. Test outcomes	17
6.2.2. Statistical criteria for measuring run_length	17
6.2.3. Classifications of tests	19
6.2.4. Reordering Tolerance	20
6.3. Test Qualifications	20
6.3.1. Verify the Traffic Generation Accuracy	20
6.3.2. Verify the absence of cross traffic	21
6.3.3. Additional test preconditions	22
7. Single Property Tests	22
7.1. Basic Data and Loss Rate Tests	22
7.1.1. Loss Rate at Paced Full Data Rate	22
7.1.2. Loss Rate at Full Data Windowed Rate	23
7.1.3. Background Loss Rate Tests	23
7.2. Standing Queue tests	24
7.2.1. Congestion Avoidance	24
7.2.2. Buffer Bloat	25
7.2.3. Duplex Self Interference	25
7.3. Slowstart tests	25
7.3.1. Full Window slowstart test	25
7.3.2. Slowstart AQM test	26
7.4. Sender Rate Burst tests	26
7.4.1. Sender TCP Send Offload (TSO) tests	26
7.4.2. Sender Full Window burst test	26
8. Combined Tests	27
8.1. Sustained burst test	27
9. Calibration	28
10. Acknowledgements	28
11. Informative References	28

Appendix A. Model Derivations	29
Appendix B. old text	29
B.1. An earlier document	30
B.2. End-to-end parameters from subpaths	31
B.3. Per subpath parameters	32
B.4. Version Control	32
Authors' Addresses	32

1. Introduction

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

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

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

Models are also used to compute the statistical criteria for evaluating the IP diagnostics tests. The criteria for passing each test must be determined from the end-to-end target performance and independent of the RTT or other properties of the subpath under test. In addition to passing or failing, a test can be inconclusive if the precomputed traffic pattern was not authentically generated, test preconditions were not met or the measurement results were not statistically significantly.

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

Section 2 defines terminology used throughout this document. It has been difficult to develop BTC metrics due to some overlooked requirements described in Section 3 and some intrinsic problems with using protocols for measurement, described in Section 4. In

Section 5 we describe the models and common parameters used to derive the targeted diagnostic suite. In Section 6 we describe common testing procedures used by all of the tests. Each subpath is evaluated using suite of far simpler and more predictable single property tests described in Section 7. Section 8 describes some combined tests that are more efficient to implement and deploy. However, if they fail they may not clearly indicate the nature of the problem.

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

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

1.1. TODO

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

Formatted: Fri Jun 21 18:23:29 PDT 2013

2. Terminology

Properties determined by the end-to-end path and application. They are described in more detail in Section 5.1.

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

Target Data Rate: The application or ultimate user's performance goal. This must be slightly smaller than the actual link rate, otherwise there is no margin for compensating for RTT or other path properties.

Target RTT (Round Trip Time): The RTT over which the application must meet the target performance.

Target MTU (Maximum Transmission Unit): Assume 1500 Bytes per packet unless otherwise specified. If some subpath forces a smaller MTU, then it becomes the target MTU, and all subpaths must be tested with the same smaller MTU.

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

Permitted Number of Connections: The target rate can be more easily obtained by dividing the traffic across more than one connection. In general the number of concurrent connections is determined by the application, however see the comments below on multiple connections.

[sender] [interface] rate: The burst data rate, constrained by the data sender's interfaces. Today 1 or 10 Gb/s are typical.

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

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

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

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

subpath Subpath as defined in [RFC2330].

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

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

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

front path The subpath from the data sender to the dominant bottleneck.

back path The subpath from the dominant bottleneck to the receiver.

return path The path taken by the ACKs from the data receiver to the data sender.

cross traffic Other, potentially interfering, traffic competing for resources (network and/or queue capacity).

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

@ @@@

pipe size The number of packets needed in flight (the window size) to exactly fill some network path or sub path. The is the window size which in normally the onset of queueing.

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

subpath pipe size

run length Observed, measured or specified number of packets that are (to be) delivered between losses or ECN marks. Nominally one over the loss probability.

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

reference_target_run_length: One specific conservative estimate of the number of packets that must be delivered between loss episodes in most diagnostic tests.

derating: The modeling framework permits some latitude in derating some specific test parameters as described in Section 5.3.

Test types [These need work]

capacity tests: For "capacity tests" is required that as long as the test traffic is within the proper envelope for the target end-to-end performance, the average packet losses must be below the threshold computed by the model.

Engineering tests: Engineering tests verify that the subpath under test interacts well with TCP style self clocked protocols using adaptive congestion control based on packet loss and ECN marks. For example "AQM Tests" verify that when the presented load exceeds the capacity of the subpath, the subpath signals for the transport protocol to slow down, by appropriately ECN marking or dropping some of the packets. Note while that cross traffic is can cause capacity tests to fail, it has the potential to cause AQM tests to false pass, which is why AQM tests require separate test procedures.

3. New requirements relative to RFC 2330

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

- o Metrics must be actionable by the ISP - they have to be interpreted in terms of behaviors or properties at the IP or lower layers, that an ISP can test, repair and verify.
- o Metrics must be vantage point invariant over a significant range of measurement point choices (e.g., measurement points as described in [I-D.morton-ippm-lmap-path]), including off path measurement points. The only requirements on MP selection should be that the portion of the path that is not under test is effectively ideal (or is non ideal in calibratable ways) and the end-to-end RTT between MPs is below some reasonable bound.
- o Metrics must be repeatable by multiple parties. It must be possible for different parties to make the same measurement and observe the same results. In particular it is specifically important that both a consumer (or their delegate) and ISP be able to perform the same measurement and get the same result.

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

4. Background

At the time the IPPM WG was chartered, sound Bulk Transport Capacity measurement was known to be beyond our capabilities. By hindsight it is now clear why it is such a hard problem:

- o TCP is a control system with circular dependencies - everything affects performance, including components that are explicitly not part of the test.
- o Congestion control is an equilibrium process, transport protocols change the network (raise loss probability and/or RTT) to conform to their behavior.
- o TCP's ability to compensate for network flaws is directly proportional to the number of roundtrips per second (i.e. inversely proportional to the RTT). As a consequence a flawed link may pass a short RTT local test even though it fails when the path is extended by a perfect network to some larger RTT.
- o TCP has a meta Heisenberg problem - Measurement and cross traffic interact in unknown and ill defined ways. The situation is actually worse than the traditional physics problem where you can at least estimate the relative momentum of the measurement and measured particles. For network measurement you can not in general determine the relative "elasticity" of the measurement traffic and cross traffic, so you can not even gage the relative magnitude of their effects on each other.

The MBM approach is to "open loop" TCP by precomputing traffic patterns that are typically generated by TCP operating at the given

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

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

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

- o The raw link rate is higher than the target data rate.
- o The raw packet loss rate is lower than required by a suitable TCP performance model
- o There is sufficient buffering at the dominant bottleneck to absorb a slowstart rate burst large enough to get the flow out of slowstart at a suitable window size.
- o There is sufficient buffering in the front path to absorb and smooth sender interface rate bursts at all scales that are likely to be generated by the application, any channel arbitration in the ACK path or other mechanisms.
- o When there is a standing queue at a bottleneck for a shared media subpath, there are suitable bounds on how the data and ACKs interact, for example due to the channel arbitration mechanism.
- o When there is a slowly rising standing queue at the bottleneck the onset of packet loss has to be at an appropriate point (time or queue depth) and progressive.

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

Note that this procedure is not invertible: a singleton measurement is a pass/fail evaluation of a given path or subpath at a given performance. Measurements to confirm that a link passes at one particular performance may not be generally be useful to predict if the link will pass at a different performance.

Although they are not invertible, they do have several other valuable properties, such as natural ways to define several different composition metrics [RFC5835].

[Add text on algebra on metrics (A-Frame from [RFC2330]) and

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

4.1. TCP properties

TCP and SCTP are self clocked protocols. The dominant steady state behavior is to have an approximately fixed quantity of data and acknowledgements (ACKs) circulating in the network. The receiver reports arriving data by returning ACKs to the data sender, the data sender most frequently responds by sending exactly the same quantity of data back into the network. The quantity of data plus the data represented by ACKs circulating in the network is referred to as the window. The mandatory congestion control algorithms incrementally adjust the widow by sending slightly more or less data in response to each ACK. The fundamentally important property of this systems is that it is entirely self clocked: The data transmissions are a reflection of the ACKs that were delivered by the network, the ACKs are a reflection of the data arriving from the network.

A number of phenomena can cause bursts of data, even in idealized networks that are modeled as simple queueing systems.

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

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

Note that if the bottleneck data rate is significantly slower than the rest of the path, the slowstart bursts will not cause significant queues anywhere else along the path; they primarily exercise the queue at the dominant bottleneck. Furthermore although the interface

rate bursts caused by the application are likely to be smaller than burst at the last RTT of slowstart, they are at a higher rate so they can exercise queues at arbitrary points along the "front path" from the data sender up to and including the queue at the bottleneck.

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

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

If a particular end-to-end path contains a link or device that alters the ACK stream, then the entire path from the sender up to the bottleneck must be tested at the burst parameters implied by the ACK scheduling algorithms. The most important parameter is the Effective Bottleneck Data Rate, which is the average rate at which the ACKs advance `snd.una`. Note that thinning the ACKs (relying on the cumulative nature of `seg.ack` to permit discarding some ACKs) is implies an effectively infinite bottleneck data rate.

To verify that a path can meet the performance target, Model Based Metrics need to independently confirm that the entire path can tolerate bursts of the dimensions that are likely to be induced by the application and any data or ACK scheduling. Two common cases are the most important: slowstart bursts of with more than the `target_pipe_size` data at twice the effective bottleneck data rate; and somewhat smaller sender interface rate bursts.

5. Common Models and Parameters

Transport performance models are used to derive the test parameters for test suites of simple diagnostics from the end-to-end target

parameters and additional ancillary parameters.

5.1. Target End-to-end parameters

The target end to end parameters are the target data rate, target RTT and target MTU as defined in Section 2. These parameters are determined by the needs of the application or the ultimate end user and the end-to-end Internet path. They are in units that make sense to the upper layer: payload bytes delivered, excluding header overheads for IP, TCP and other protocol.

Ancillary parameters include the effective bottleneck rate and the permitted number of connections (`numb_cons`).

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

5.2. Common Model Calculations

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

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

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

The `reference_target_run_length`, which is the most conservative model for the minimum spacing between losses, can be derived as follows: assume the `link_data_rate` is infinitesimally larger than the `target_data_rate`. Then `target_pipe_size` also predicts the onset of queueing. If the transport protocol (e.g. TCP) has an average window size that is larger than the `target_pipe_size`, the excess packets will form a standing queue at the bottleneck.

If the transport protocol is using standard Reno style Additive Increase, Multiplicative Decrease congestion control [RFC5681], then

there must be `target_pipe_size` roundtrips between losses. Otherwise the multiplicative window reduction triggered by a loss would cause the network to be underfilled. Following [MSM097], we derive the losses must be no more frequent than every 1 in $(3/2)(\text{target_pipe_size}^2)$ packets. This provides the reference value for `target_run_length` which is typically the number of packets that must be delivered between loss episodes in the tests below:

```
reference_target_run_length = (3/2)(target_pipe_size^2)
```

Note that this calculation is based on a number of assumptions that may not apply. Appendix A discusses these assumptions and provides some alternative models. The actual method for computing `target_run_length` MUST be documented along with the rationale for the underlying assumptions and the ratio of chosen `target_run_length` to `reference_target_run_length`. @@@ MOVE

Although this document gives a lot of latitude for calculating `target_run_length`, people designing suites of tests need to consider the effect of their choices on the ongoing conversation and tussle about the relevance of "TCP friendliness" as an appropriate model for capacity allocation. Choosing a `target_run_length` that is substantially smaller than `reference_target_run_length` is equivalent to saying that it is appropriate for the transport research community to abandon "TCP friendliness" as a fairness model and to develop more aggressive Internet transport protocols, and for applications to continue (or even increase) the number of connections that they open concurrently.

The calculations for individual parameters are presented with the each single property test. In general these calculations permit some derating as described in Section 5.3. For test parameters that can be derated and are proportional to `target_pipe_size`, it is recommended that the derating be specified relative to `target_pipe_size` calculations using `numb_cons=1`, although the derating may additionally be specified relative to the `target_pipe_size` common to other tests.

5.3. Parameter Derating

Since some aspects of the models are very conservative, the modeling framework permits some latitude in derating some specific test parameters. For example classical performance models suggest that in order to be sure that a single TCP stream can fill a link, it needs to have a full bandwidth-delay-product worth of buffering at the bottleneck[QueueSize]. In real networks with real applications this is often overly conservative. Rather than trying to formalize more complicated models we permit some test parameters to be relaxed as

long as they meet some additional procedural constraints:

- o The method used compute and justify the derated metrics is published in such a way that it becomes a matter of public record. @@@ introduce earlier
- o The calibration procedures described in Section 9 are used to demonstrate the feasibility of meeting the performance targets with the derated test parameters.
- o The calibration process itself is documented in such a way that other researchers can duplicate the experiments and validate the results.

In the test specifications in Section 7 assume $0 < \text{derate} \leq 1$, is a derating parameter. These will be individually named in the final document. In all cases making derate smaller makes the test more tolerant. Derate = 1 is "full strenght".

Note that some test parameters are not permitted to be derated.

6. Common testing procedures

6.1. Traffic generating techniques

6.1.1. Paced transmission

Paced (burst) transmissions: send bursts of data on a timer to meet a particular target rate and pattern.

Single: Send individual packets at the specified rate or headway.

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

Slowstart: Send 4 packet sender interface rate bursts at an average rate equal to the minimum of twice effective bottleneck link rate or the sender interface rate. This corresponds to the average rate during a TCP slowstart when Appropriate Byte Counting [ABC] is present or delayed ack is disabled.

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

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

6.1.2. Constant window pseudo CBR

Implement pseudo CBR by running a standard protocol such as TCP with a fixed window size. This has the advantage that it can be implemented as part of real content delivery. The rate is only maintained in average over each RTT, and is subject to limitations of the transport protocol.

For tests that have strongly prescribed data rates, if the transport protocol fails to maintain the test rate for any reason related to the network itself, such as packet losses or congestion, the test should be considered inconclusive. Otherwise there are some cases where tester failures might cause false negative link test results.

6.1.2.1. Scanned window pseudo CBR

Same as the above, except the window is incremented once per $2 * \text{target_pipe_size}$, starting from below `target_pipe[@@@ test pipe]` and sweeping up to first loss or some other event. This is analogous to the tests implemented in Windowed Ping [WPING] and pathdiag [Pathdiag]

6.1.3. Intermittent Testing

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

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

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

6.1.4. Intermittent Scatter Testing

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

6.2. Interpreting the Results

6.2.1. Test outcomes

A singleton is a pass fail measurement. If any subpath fails any test it can be assumed that the end-to-end path will also fail to attain the target performance under some conditions.

In addition we use "inconclusive" outcome to indicate that a test failed to attain the required test conditions. This is important to the extent that the tests themselves use protocols that have built in control systems which might interfere with some aspect of the test. For example consider a test is implemented by adding rate controls and instrumentation to TCP: failing to attain the specified data rate has to be treated as inconclusive, unless the test clearly fails (target_run_length is too small). This is because failing to reach the target rate is an ambiguous signature for problems with either the test procedure (a problem with the TCP implementation or the test path RTT is too long) or the subpath itself.

The vantage independence properties of Model Based Metrics depends on the accuracy of the distinction between failing and inconclusive tests. One of the goals of evolving test designs will be to keep sharpening the distinction between failing and inconclusive tests.

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

6.2.2. Statistical criteria for measuring run_length

When evaluating the observed run_length, we need to determine appropriate packet stream sizes and acceptable error levels to test efficiently. In practice, can we compare the empirically estimated loss probabilities with the targets as the sample size grows? How large a sample is needed to say that the measurements of packet transfer indicate a particular run-length is present?

The generalized measurement can be described as recursive testing:

send a flight of packets and observe the packet transfer performance (loss ratio or other metric, any defect we define).

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

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

H0: no more than one defects in `target_run_length = (3/2)*(flight)^2` packets

and we can stop sending flights of packets if measurements support accepting H0 with the specified Type I error = α (= 0.05 for example).

We also have an alternative Hypothesis to evaluate: if performance is significantly lower than the `target_defect_probability`, say half the target:

H1: one or more defects in `target_run_length/2` packets

and we can stop sending flights of packets if measurements support rejecting H0 with the specified Type II error = β , thus preferring the alternate H1.

H0 and H1 constitute the Success and Failure outcomes described elsewhere in the memo, and while the ongoing measurements do not support either hypothesis the current status of measurements is inconclusive.

The problem above is formulated to match the Sequential Probability Ratio Test (SPRT) [StatQC] [temp ref: http://en.wikipedia.org/wiki/Sequential_probability_ratio_test], which also starts with a pair of hypothesis specified as above:

H0: $p = p_0$ = one defect in `target_run_length`

H1: $p = p_1$ = one defect in `target_run_length/2`

As flights are sent and measurements collected, the tester evaluates the cumulative log-likelihood ratio:

$S_i = S_{i-1} + \log(\text{Lambda}_i)$

where `Lambda_i` is the ratio of the two likelihood functions (calculated on the measurement at packet `i`, and index `i` increases

linearly over all flights of packets) for p0 and p1 [temp ref:
http://en.wikipedia.org/wiki/Likelihood_function].

The SPRT specifies simple stopping rules:

- o $a < S_i < b$: continue testing
- o $S_i \leq a$: Accept H0
- o $S_i \geq b$: Accept H1

where a and b are based on the Type I and II errors, alpha and beta:

$a \approx \text{Log}((\beta/(1-\alpha)))$ and $b \approx \text{Log}((1-\beta)/\alpha)$

with the error probabilities decided beforehand, as above.

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

6.2.3. Classifications of tests

Tests are annotated with "(capacity)", "(engineering)" or "(monitoring)". @@@@MOVE to definitions?

Capacity tests determine if a network subpath has sufficient capacity to deliver the target performance. As such, they reflect parameters that can transition from passing to failing as a consequence of additional presented load or the actions of other network users. By definition, capacity tests also consume network resources (capacity and/or buffer space), and their test schedules must be balanced by their cost.

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

Engineering tests evaluate how network algorithms (such as AQM and channel allocation) interact with transport protocols. These tests are likely to have complicated interactions with other network traffic and can be inversely sensitive to load. For example a test to verify that an AQM algorithm causes ECN marks or packet drops early enough to limit queue occupancy may experience a false pass results in the presence of bursty cross traffic. It is important that engineering tests be performed under a wide range of conditions, including both in situ and bench testing, and under a variety of load conditions. Ongoing monitoring is less likely to be useful for these tests, although sparse in situ testing might be appropriate.

@@@ Add single property vs combined tests here?

6.2.4. Reordering Tolerance

All tests must be instrumented for reordering [RFC4737].

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

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

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

NB: Current TCP implementations are not compatible with this metric. We view this as bugs in current TCP implementations.

Parameters:

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

6.3. Test Qualifications

Things to monitor before, during and after a test.

6.3.1. Verify the Traffic Generation Accuracy

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

Parameters:

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

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

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

6.3.2. Verify the absence of cross traffic

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

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

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

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

Parameters:

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

One possible method is an adaptation of: [www-didc.lbl.gov/papers/SCNM-PAM03.pdf](http://www.didc.lbl.gov/papers/SCNM-PAM03.pdf) D Agarwal et al. "An Infrastructure for Passive Network Monitoring of Application Data Streams". Use the same technique as that paper to trigger the capture of SNMP statistics for the link.

6.3.3. Additional test preconditions

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

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

7. Single Property Tests

7.1. Basic Data and Loss Rate Tests

We propose several versions of the loss rate test. All are rate controlled at or below the `target_data_rate`. The first, performed at constant full data rate, is intrusive and recommend for infrequent testing, such as when a service is first turned up or as part of an auditing process. The second, background loss rate, is designed for ongoing monitoring for change in subpath quality.

7.1.1. Loss Rate at Paced Full Data Rate

Confirm that the observed run length is at least the `target_run_lenght` while sending at the `target_rate`. This test implicitly confirms that `sub_path` has sufficient raw capacity to carry the `target_data_rate`. This version of the loss rate test relies on timers to schedule data transmission at a true constant bit rate (CBR).

Test Parameters:

Run Length Same as `target_run_lenght`

Data Rate Same as `target_data_rate`

Maximum Cross Traffic A specified small fraction of `target_data_rate`.

Note that `target_run_lenght` and `target_data_rate` parameters MUST NOT be derated. If the default parameters are too stringent an alternate model as described in Appendix A can be used to compute `target_run_lenght`.

The test traffic is sent using the procedures in Section 6.1.1 at `target_data_rate` with a burst size of 1, subject to the qualifications in Section 6.3. The receiver accumulates packet delivery statistics as described in Section 6.2 to score the outcome:

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

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

Inconclusive: The test failed to meet the qualifications defined in Section 6.3 or neither test was statistically significant.

7.1.2. Loss Rate at Full Data Windowed Rate

Confirm that the observed run length is at least the `target_run_lenght` while sending at the `target_rate`. This test implicitly confirms that `sub_path` has sufficient raw capacity to carry the `target_data_rate`. This version of the loss rate test relies on a fixed window to self clock data transmission into the network. This is more authentic.

Test Parameters:

Run Length Same as `target_run_lenght`

Data Rate Same as `target_data_rate`

Maximum Cross Traffic A specified small fraction of `target_data_rate`.

Note that `target_run_lenght` and `target_data_rate` parameters MUST NOT be derated. If the default parameters are too stringent an alternate model as described in Appendix A can be used to compute `target_run_lenght`.

The test traffic is sent using the procedures in Section 6.1.1 at `target_data_rate` with a burst size of 1, subject to the qualifications in Section 6.3. The receiver accumulates packet delivery statistics as described in Section 6.2 to score the outcome:

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

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

Inconclusive: The test failed to meet the qualifications defined in Section 6.3 or neither test was statistically significant.

7.1.3. Background Loss Rate Tests

The background loss rate is a low rate version of the target rate test above, designed for ongoing monitoring for changes in subpath quality without disrupting users. It should be used in conjunction with the above full rate test because it may be subject to false results under some conditions, in particular it may false pass changes in underlying link properties (e.g. a modem retraining to an

out of contract lower rate).

Parameters:

Run Length Same as target_run_length

Data Rate Some small fraction of target_data_rate, such as 1%.

Once the preconditions described in Section 6.3 are met, the test data is sent at the prescribed rate with a burst size of 1. The receiver accumulates packet delivery statistics and the procedures described in Section 6.2.1 and Section 6.3 are used to score the outcome:

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

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

Inconclusive: Neither test was statistically significant or there was excess cross traffic during the test.

7.2. Standing Queue tests

These tests confirm that the bottleneck is well behaved across the onset of queueing. For conventional bottlenecks this will be from the onset of queuing to the point where there is a full target_pipe of standing data. Well behaved generally means lossless for target_run_length, followed by a small number of losses to signal to the transport protocol that it should slow down. Losses that are too early can prevent the transport from averaging above the target_rate. Losses that are too late indicate that the queue might be subject to bufferbloat and subject other flows to excess queuing delay. Excess losses (more than half of target_pipe) make loss recovery problematic for the transport protocol.

These tests can also observe some problems with channel acquisition systems, especially at the onset of persistent queueing. Details TBD.

7.2.1. Congestion Avoidance

Use the procedure in Section 6.1.2.1 to sweep the window (rate) from below link_pipe up to beyond target_pipe+link_pipe. Depending on events that happen during the scan, score the link. Identify the power_point=MAX(rate/RTT) as the start of the test.

Fail if first loss is too early (loss rate too high) on repeated tests or if the losses are more than half of the outstanding data. (a

capacity test)

7.2.2. Buffer Bloat

Use the procedure in Section 6.1.2.1 to sweep the window (rate) from below `link_pipe` up to beyond `target_pipe+link_pipe`. Depending on events that happen during the scan, score the link. Identify the "power point:MAX(rate/RTT) as the start of the test (should be `window=target_pipe`)

Fail if first loss is too late (insufficient AQM and subject to `bufferbloat` - an engineering test). NO THEORY

7.2.3. Duplex Self Interference

Use the procedure in Section 6.1.2.1 to sweep the window (rate) from below `link_pipe` up to beyond `target_pipe+required_queue`. Depending on events that happen during the scan, score the link. Identify the "power point:MAX(rate/RTT) as the start of the test (should be `window=target_pipe`) @@@ add `required_queue` and `power_point`

Fail if RTT is non-monotonic by more than a small number of packet times (channel allocation self interference - engineering) IS THIS SUFFICIENT?

7.3. Slowstart tests

These tests mimic slowstart: data is sent at `slowstart_rate` (twice `subpath_rate`). They are deemed inconclusive if the elapsed time to send the data burst is not less than half of the (extrapolated) time to receive the ACKs. (i.e. sending data too fast is ok, but sending it slower than twice the actual bottleneck rate is deemed inconclusive). Space the bursts such that the average ACK rate is equal to or faster than the `target_data_rate`.

These tests are not useful at burst sizes smaller than the sender interface rate tests, since the sender interface rate tests are more strenuous. If it is necessary to derate the sender interface rate tests, then the full window slowstart test (un-derated) would be important.

7.3.1. Full Window slowstart test

Send $(\text{target_pipe_size} + \text{required_queue}) * \text{derate}$ bursts must have fewer than one loss per $\text{target_run_length} * \text{derate}$. Note that these are the same parameters as the Sender Full Window burst test, except the burst rate is at slowstart rate, rather than sender interface rate. SHOULD `derate=1`.

Otherwise TCP will exit from slowstart prematurely, and only reach a full `target_pipe_size` window by way of congestion avoidance.

This is a capacity test: cross traffic may cause premature losses.

7.3.2. Slowstart AQM test

Do a continuous slowstart (`data rate = slowstart_rate`), until first loss, and repeat, gathering statistics on the last delivered packet's RTT and window size. Fail if too large (NO THEORY for value).

This is an engineering test: It would be best performed on a quiescent network or testbed, since cross traffic might cause a false pass.

7.4. Sender Rate Burst tests

These tests use "sender interface rate" bursts. Although this is not well defined it should be assumed to be current state of the art server grade hardware (often 10Gb/s today). (load)

7.4.1. Sender TCP Send Offload (TSO) tests

If $\text{MIN}(\text{target_pipe_size}, 42)$ packet bursts meet `target_run_lenght` (Not derated!).

Otherwise the link will interact badly with modern server NIC implementations, which as an optimization to reduce host side interactions (interrupts etc) accept up to 64kB super packets and send them as 42 separate packets on the wire side.cc (load)

7.4.2. Sender Full Window burst test

`target_pipe_size*derate` bursts have fewer than one loss per `target_run_length*derate`.

Otherwise application pauses will cause unwarranted losses. Current standards permit TCP to send a full `cwnd` burst following an application pause. (`Cwnd` validation is not required, but even so does not take effect until the pause is longer than `RTT`).

NB: there is no model here for what is good enough. `derate=1` is safest, but may be unnecessarily conservative for some applications. Some application, such as streaming video need `derate=1` to be efficient when the application pacing quanta is larger than `cwnd`. (load)

8. Combined Tests

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

8.1. Sustained burst test

Send `target_pipe_size` sender interface rate bursts every `target_RTT`, verify that the observed run length meets `target_run_length`. Key observations:

- o This test is RTT invariant, as long as the tester can generate the required pattern.
- o The subpath under test is expected to go idle for some fraction of the time: $(\text{link_rate} - \text{target_rate}) / \text{link_rate}$. Failing to do so suggests a problem with the procedure.
- o This test is more strenuous than the slowstart tests: they are not needed if the link passes underated sender interface rate burst tests.
- o This test could be derated by reducing both the burst size and headway (same average data rate).
- o A link that passes this test is likely to be able to sustain higher rates (close to `link_rate`) for paths with RTTs smaller than the `target_RTT`. Offsetting this performance underestimation is the rationale behind permitting derating in general.
- o This test should be implementable with standard instrumented TCP, [RFC 4898] using a specialized measurement application at one end and a minimal service at the other end [RFC 863, RFC 864]. It may require tweaks to the TCP implementation.
- o This test is efficient to implement, since it does not require per-packet timers, and can make maximal use of TSO in modern NIC hardware.
- o This test is not totally sufficient: the standing window engineering tests are also needed to be sure that the link is well behaved at and beyond the onset of congestion.
- o I believe that this test can be proven to be the one capacity test to supplant them all.

Example

To confirm that a 100 Mb/s link can reliably deliver single 10 MByte/s stream at a distance of 50 mS, test the link by sending 346 packet bursts every 50 mS (10 MByte/s payload rate, assuming a 1500 Byte IP MTU and 52 Byte TCP/IP headers). These bursts are 4196288 bits on the wire (assuming 16 bytes of link overhead and framing) for an aggregate test data rate of 8.4 Mb/s.

To pass the test using the most conservative TCP model for a single stream the observed run length must be larger than 179574 packets.

This is the same as less than one loss per 519 bursts (1.5×346) or every 26 seconds.

Note that this test potentially cause transient 346 packet queues at the bottleneck.

9. Calibration

If using derated metrics, or when something goes wrong, the results must be calibrated against a traditional BTC. The preferred diagnostic follow-up to calibration issues is to run open end-to-end measurements on an open platform, such as Measurement Lab [<http://www.measurementlab.net/>]

10. Acknowledgements

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

Meredith Whittaker for improving the clarity of the communications.

11. Informative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.
- [I-D.morton-ippm-lmap-path] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A Reference Path and Measurement Points for LMAP", draft-morton-ippm-lmap-path-00 (work in progress), January 2013.

- [MSMO97] Mathis, M., Semke, J., Mahdavi, J., and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review volume 27, number3, July 1997.
- [WPING] Mathis, M., "Windowed Ping: An IP Level Performance Diagnostic", INET 94, June 1994.
- [Pathdiag] Mathis, M., Heffner, J., O'Neil, P., and P. Siemsen, "Pathdiag: Automated TCP Diagnosis", Passive and Active Measurement , June 2008.
- [BScope] Browserscope, "Browserscope Network tests", Sept 2012, <<http://www.browserscope.org/?category=network>>.
- [Rtool] R Development Core Team, "R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>", , 2011.
- [StatQC] Montgomery, D., "Introduction to Statistical Quality Control - 2nd ed.", ISBN 0-471-51988-X, 1990.
- [CVST] Krueger, T. and M. Braun, "R package: Fast Cross-Validation via Sequential Testing", version 0.1, 11 2012.

Appendix A. Model Derivations

This appendix describes several different ways to calculate `target_run_length` and the implication of the chosen calculation.

Rederive MSMO97 under two different assumptions: `target_rate = link_rate` and `target_rate < 2 * link_rate`.

Show equivalent derivation for CUBIC.

Commentary on the consequence of the choice.

Appendix B. old text

This entire section is contains scraps of text to be moved, removed or absorbed elsewhere in the document

B.1. An earlier document

Step 0: select target end-to-end parameters: a target rate and target RTT. The primary test will be to confirm that the link quality is sufficient to meet the specified target rate for the link under test, when extended to the target RTT by an ideal network. The target rate must be below the actual link rate and nominally the target RTT would be longer than the link RTT. There should probably be a convention for the relationship between link and target rates (e.g. 85%).

For example on a 10 Mb/s link, the target rate might be 1 MBytes/s, at an RTT of 100 ms (a typical continental scale path).

Step 1: On the basis of the target rate and RTT and your favorite TCP performance model, compute the "required run length", which is the required number of consecutive non-losses between loss episodes. The run length resembles one over the loss probability, if clustered losses only count as a single event. Also select "test duration" and "test rate". The latter would nominally be the same as the target rate, but might be different in some situations. There must be documentation connecting the test rate, duration and required run length, to the target rate and RTT selected in step 0.

Continuing the above example: Assuming a 1500 Byte MTU. The calculated model loss rate for a single TCP stream is about 0.01% (1 loss in 1E4 packets).

Step 2, the actual measurement proceeds as follows: Start an unconstrained bulk data flow using any modern TCP (with large buffers and/or autotuning). During the first interval (no rate limits) observe the slowstart (e.g. tcpdump) and measure: Peak burst size; link clock rate (delivery rate for each round); peak data rate for the fastest single RTT interval; fraction of segments lost at the end of slowstart. After the flow has fully recovered from the slowstart (details not important) throttle the flow down to the test rate (by clamping cwnd or application pacing at the sender or receiver). While clamped to the test rate, observe the losses (run length) for the chosen test duration. The link passes the test if the slowstart ends with less than approximately 50% losses and no timeouts, the peak rate is at least the target rate, and the measured run length is better than the required run length. There will also need to be some ancillary metrics, for example to discard tests where the receiver closes the window, invalidating the slowstart test. [This needs to be separated into multiple subtests]

Optional step 3: In some cases it might make sense to compute an "extrapolated rate", which is the minimum of the observed peak rate, and the rate computed from the specified target RTT and the observed

run length by using a suitable TCP performance model. The extrapolated rate should be annotated to indicate if it was run length or peak rate limited, since these have different predictive values.

Other issues:

If the link RTT is not substantially smaller than the target RTT and the actual run length is close to the target rate, a standards compliant TCP implementation might not be effective at accurately controlling the data rate. To be independent of the details of the TCP implementation, failing to control the rate has to be treated as a spoiled measurement, not a infrastructure failure. This can be overcome by "stiffening" TCP by using a non-standard congestion control algorithm. For example if the rate controlling by clamping cwnd then use "relentless TCP" style reductions on loss, and lock ssthresh to the cwnd clamp. Alternatively, implement an explicit rate controller for TCP. In either case the test must be abandoned (aborted) if the measured run length is substantially below the target run length.

If the test is run "in situ" in a production environment, there also needs to be baseline tests using alternate paths to confirm that there are no bottlenecks or congested links between the test end points and the link under test.

It might make sense to run multiple tests with different parameters, for example infrequent tests with test rate equal to the target rate, and more frequent, less disruptive tests with the same target rate but the test rate equal to 1% of the target rate. To observe the required run length, the low rate test would take 100 times longer to run.

Returning to the example: a full rate test would entail sending 690 pps (1 MByte/s) for several tens of seconds (e.g. 50k packets), and observing that the total loss rate is below 1:1e4. A less disruptive test might be to send at 6.9 pps for 100 times longer, and observing

B.2. End-to-end parameters from subpaths

[This entire section needs to be overhauled and should be skipped on a first reading. The concepts defined here are not used elsewhere.]

The following optional parameters apply for testing generalized end-to-end paths that include subpaths with known specific types of behaviors that are not well represented by simple queueing models:

Bottleneck link clock rate: This applies to links that are using virtual queues or other techniques to police or shape users traffic at lower rates full link rate. The bottleneck link clock rate should be representative of queue drain times for short bursts of packets on an otherwise unloaded link.

Channel hold time: For channels that have relatively expensive channel arbitration algorithms, this is the typical (maximum?) time that data and or ACKs are held pending acquiring the channel. While under heavy load, the RTT may be inflated by this parameter, unless it is built into the target RTT

Preload traffic volume: If the user's traffic is shaped on the basis of average traffic volume, this is volume necessary to invoke "heavy hitter" policies.

Unloaded traffic volume: If the user's traffic is shaped on the basis of average traffic volume, this is the maximum traffic volume that a test can use and stay within a "light user" policies.

Note on a ConEx enabled network [ConEx], the word "traffic" in the last two items should be replaced by "congestion" i.e. "preload congestion volume" and "unloaded congestion volume".

B.3. Per subpath parameters

[This entire section needs to be overhauled and should be skipped on a first reading. The concepts defined here are not used elsewhere.]

Some single parameter tests also need parameter of the subpath.

subpath RTT: RTT of the subpath under test.

subpath link clock rate: If different than the Bottleneck link clock rate

B.4. Version Control

Formatted: Fri Jun 21 18:23:29 PDT 2013

Authors' Addresses

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

Email: mattmathis@google.com

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

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

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

A. Morton
AT&T Labs
February 1, 2013

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

Abstract

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

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 5, 2013.

Copyright Notice

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

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

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

Table of Contents

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

1. Introduction

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

The access-rate scenario or use case has wide-spread attention of Internet access subscribers and seemingly all Internet industry players, including regulators. This problem is being approached with many different measurement methods. This memo

2. Purpose and Scope

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

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

We characterize the access rate measurement scenario as follows:

- o The Access portion of the network is the focus of this problem statement. The user typically subscribes to a service with bi-directional access partly described by rates in bits per second. The rates may be expressed as raw capacity or restricted capacity as described in [RFC6703]. These are the quantities that must be measured according to one or more standard metrics for which methods must also be agreed as a part of the solution.
- o Referring to the reference path defined in [I-D.morton-ippm-lmap-path], possible measurement points include a Subscriber's host (mp000), the access service demarcation point (mp100), Intra IP access where a globally routable address is present (mp150), or the gateway between the measured access network and other networks (mp190).
- o Rates at the edge of the network are several orders of magnitude less than aggregation and core portions.
- o Asymmetrical ingress and egress rates are prevalent.

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

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

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

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

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

The actual path used by traffic may influence the rate measurement results for some forms of access, as it may differ between user and test traffic if the test traffic has different characteristics, primarily in terms of the packets themselves (the Type-P described in [RFC2330]).

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

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

- o Transport Protocol port numbers used

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

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

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

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

It is a non-goal to standardize methods of measurement in this memo. However, the problem statement will mandate that support for one or more categories of rate measurement methods and adequate control features for the methods in the test protocol.

3. Active Rate Measurement

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

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

Most forms of active testing intrude on user performance to some degree. One key tenet of IPPM methods is to minimize test traffic effects on user traffic in the production network. Section 5 of

[RFC2680] lists the problems with high measurement traffic rates, and the most relevant for rate measurement is the tendency for measurement traffic to skew the results, followed by the possibility of introducing congestion on the access link. Obviously, categories of rate measurement methods that use less active test traffic than others with similar accuracy SHALL be preferred for In-Service testing.

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

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

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

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

SENDER:

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

RECEIVER:

Collect streams of test packets with various characteristics (as

described above), and make the measurements necessary to support rate measurement at the other end of an end-user access or aggregated leased network segment.

REPORTER:

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

4. Measurement Method Categories

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

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

1. Packet pairs, with fixed intra-pair packet spacing and fixed or random time intervals between pairs in a test stream.
2. Multiple streams of packet pairs, with a range of intra-pair spacing and inter-pair intervals.
3. One or more packet ensembles in a test stream, using a fixed ensemble size in packets and one or more fixed intra-ensemble packet spacings (including zero spacing).
4. One or more packet chirps, where intra-packet spacing typically decreases between adjacent packets in the same chirp and each pair of packets represents a rate for testing purposes.

For all categories, the test protocol MUST support:

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

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

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

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

>>>>>>

Note: For measurement systems employing TCP Transport protocol, the ability to generate specific stream characteristics requires a sender with the ability to establish and prime the connection such that the desired stream characteristics are allowed. See Mathis' work in progress for more background [I-D.mathis-ippm-model-based-metrics]. The general requirement statements needed to describe an "open-loop" TCP sender require some additional discussion.

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

>>>>>>

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

5. Test Protocol Control & Generation Requirements

Essentially, the test protocol MUST support the measurement features described in the sections above. This requires:

1. Communicating all test variables to the Sender and Receiver
2. Results collection in a one-way architecture
3. Remote device control for both one-way and two-way architectures

4. Asymmetric and/or pseudo-one-way test capability in a two-way measurement architecture

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

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

6. Security Considerations

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

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

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

Dave McDysan provided comments and text for the aggregated leased use case. Yaakov Stein suggested many considerations to address, including the In-Service vs. Out-of-Service distinction and its implication on test traffic limits and protocols. Bill Cervený and Marcelo Bagnulo have contributed insightful, clarifying comments that made this a better draft.

9. Appendix

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

10. References

10.1. Normative References

- [RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, August 2009.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, August 2010.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, October 2010.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

10.2. Informative References

- [I-D.mathis-ippm-model-based-metrics]
Mathis, M., "Model Based Internet Performance Metrics",

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

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

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

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

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

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

Author's Address

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

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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 21, 2013

L. Ciavattone
AT&T Labs
R. Geib
Deutsche Telekom
A. Morton
AT&T Labs
M. Wieser
Technical University Darmstadt
February 17, 2013

Test Plan and Results for Advancing RFC 2680 on the Standards Track
draft-ietf-ippm-testplan-rfc2680-02

Abstract

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

In this version, the results are presented in the R-tool output form. Beautification is future work.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 21, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	4
1.1. RFC 2680 Coverage	5
2. A Definition-centric metric advancement process	5
3. Test configuration	6
4. Error Calibration, RFC 2680	10
4.1. Clock Synchronization Calibration	10
4.2. Packet Loss Determination Error	10
5. Pre-determined Limits on Equivalence	11
6. Tests to evaluate RFC 2680 Specifications	12
6.1. One-way Loss, ADK Sample Comparison	12
6.1.1. 340B/Periodic Cross-imp. results	13
6.1.2. 64B/Periodic Cross-imp. results	14
6.1.3. 64B/Poisson Cross-imp. results	15
6.1.4. Conclusions on the ADK Results for One-way Packet Loss	16
6.2. One-way Loss, Delay threshold	16
6.2.1. NetProbe results for Loss Threshold	17
6.2.2. Perfas Results for Loss Threshold	18
6.2.3. Conclusions for Loss Threshold	18
6.3. One-way Loss with Out-of-Order Arrival	18
6.4. Poisson Sending Process Evaluation	19
6.4.1. NetProbe Results	20
6.4.2. Perfas Results	21
6.4.3. Conclusions for Goodness-of-Fit	23
6.5. Implementation of Statistics for One-way Delay	23
7. Conclusions for RFC 2680bis	23
8. Security Considerations	24
9. IANA Considerations	24
10. Acknowledgements	24
11. References	24
11.1. Normative References	24
11.2. Informative References	26
Authors' Addresses	26

1. Introduction

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

A renewed work effort sought to investigate ways in which the measurement variability could be reduced and thereby simplify the problem of comparison for equivalence.

There is consensus [RFC6576] that the metric definitions should be the primary focus of evaluation rather than the implementations of metrics, and equivalent results are deemed to be evidence that the metric specifications are clear and unambiguous. This is the metric specification equivalent of protocol interoperability. The advancement process either produces confidence that the metric definitions and supporting material are clearly worded and unambiguous, OR, identifies ways in which the metric definitions should be revised to achieve clarity.

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

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

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

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

Although the conclusion reached through testing is that [RFC2680] should be advanced on the Standards Track with modifications, the revised text of RFC 2680bis is not yet ready for review. Therefore, this memo documents the information to support [RFC2680] advancement,

and the approval of RFC2680bis is left for future action.

1.1. RFC 2680 Coverage

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

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

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

2. A Definition-centric metric advancement process

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

IF two implementations do not measure an equivalent singleton or sample, or produce the an equivalent statistic,

AND sources of measurement error do not adequately explain the lack of agreement,

THEN the details of each implementation should be audited along with the exact definition text, to determine if there is a lack of clarity that has caused the implementations to vary in a way that affects the correspondence of the results.

IF there was a lack of clarity or multiple legitimate interpretations of the definition text,

THEN the text should be modified and the resulting memo proposed for consensus and advancement along the standards track.

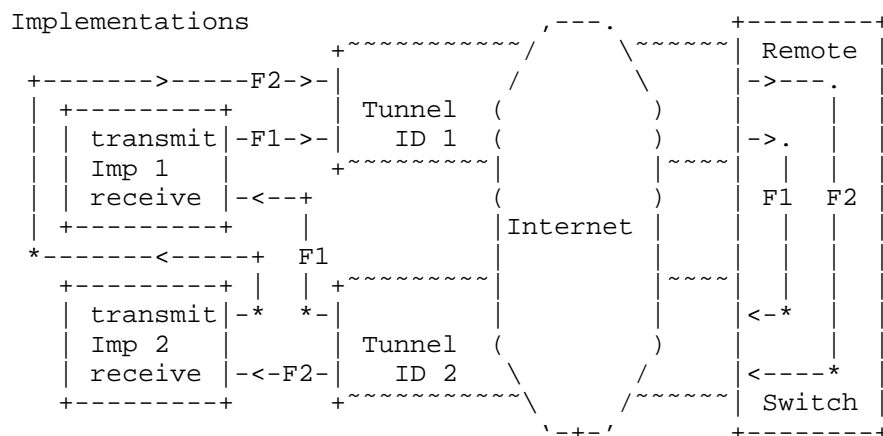
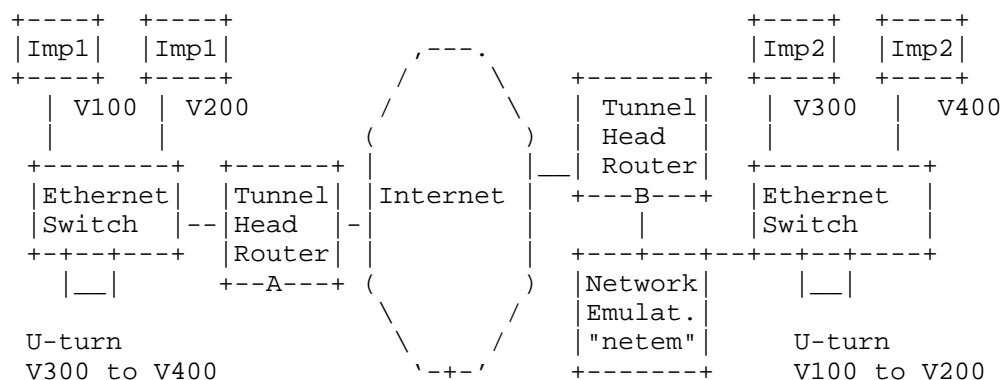
Finally, all the findings MUST be documented in a report that can support advancement on the standards track, similar to those described in [RFC5657]. The list of measurement devices used in testing satisfies the implementation requirement, while the test results provide information on the quality of each specification in the metric RFC (the surrogate for feature interoperability).

3. Test configuration

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

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

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



Illustrations of a test setup with a bi-directional tunnel. The upper diagram emphasizes the VLAN connectivity and geographical location. The lower diagram shows example flows traveling between two measurement implementations (for simplicity, only two flows are shown).

Figure 1

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

At each end of the tunnel, one pair of VLANs encapsulated in the

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

The network emulator is a host running Fedora 14 Linux [<http://fedoraproject.org/>] with IP forwarding enabled and the "netem" Network emulator as part of the Fedora Kernel 2.6.35.11 [<http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>] loaded and operating. Connectivity across the netem/Fedora host was accomplished by bridging Ethernet VLAN interfaces together with "brctl" commands (e.g., eth1.100 <-> eth2.100). The netem emulator was activated on one interface (eth1) and only operates on test streams traveling in one direction. In some tests, independent netem instances operated separately on each VLAN.

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

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

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

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

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

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

With (Section 3.2. [RFC2680]) Metric Parameters:

- + Src, the IP address of a host (12.3.167.16 or 193.159.144.8)
- + Dst, the IP address of a host (193.159.144.8 or 12.3.167.16)
- + T0, a time

- + Tf, a time
- + lambda, a rate in reciprocal seconds
- + Thresh, a maximum waiting time in seconds (see Section 2.8.2 of [RFC2680]) and (Section 3.8. [RFC2680])

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

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

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

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

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

Perfas+ does not support traceroute.

```
IPLGW#traceroute 193.159.144.8
```

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

```
Tracing the route to 193.159.144.8
```

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

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

4. Error Calibration, RFC 2680

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

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

4.1. Clock Synchronization Calibration

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

4.2. Packet Loss Determination Error

Since both measurement implementations have resource limitations, it is theoretically possible that these limits could be exceeded and a

packet that arrived at the destination successfully might be discarded in error.

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

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

5. Pre-determined Limits on Equivalence

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

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

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

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

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

For Anderson-Darling Goodness-of-Fit (ADGoF) [Radgof] comparisons, the required level of significance for the same-implementation Goodness-of-Fit (GoF) SHALL be 0.05 or 5%, as specified in Section 11.4 of [RFC2330]. This is equivalent to a 95% confidence factor.

6. Tests to evaluate RFC 2680 Specifications

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

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

6.1. One-way Loss, ADK Sample Comparison

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

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

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

1. Configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.
2. Measure a sample of one-way packet loss singletons with 2 or more implementations, using identical options and network emulator settings (if used).
3. Measure a sample of one-way packet loss singletons with *four or more* instances of the *same* implementations, using identical options, noting that connectivity differences SHOULD be the same as for the cross implementation testing.
4. If less than ten test streams are available, skip to step 7.
5. Apply the ADK comparison procedures (see Appendix C of [RFC6576]) and determine the resolution and confidence factor for

distribution equivalence of each same-implementation comparison and each cross-implementation comparison.

6. Take the coarsest resolution and confidence factor for distribution equivalence from the same-implementation pairs, or the limit defined in Section 5 above, as a limit on the equivalence threshold for these experimental conditions.
7. Compare the cross-implementation ADK performance with the equivalence threshold determined in step 5 to determine if equivalence can be declared.

The common parameters used for tests in this section are:

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

In the result analysis of this section:

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

6.1.1. 340B/Periodic Cross-imp. results

Tests described in this section used:

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

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

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

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

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

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

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

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

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

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

6.1.2. 64B/Periodic Cross-imp. results

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

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

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

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

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

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

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

Null Hypothesis: All samples come from a common population.

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

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

The cross-implementation comparisons pass the ADK criterion.

6.1.3. 64B/Poisson Cross-imp. results

Tests described in this section used:

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

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

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

```
A27cross_loss_ADK <- adk.test(A27aps_loss, A27per_loss)
```

```
> A27cross_loss_ADK
Anderson-Darling k-sample test.
```

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

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

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

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

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

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

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

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

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

6.2. One-way Loss, Delay threshold

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

See Section 2.8.2 of [RFC2680].

1. configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.
2. configure the network emulator to add 1.0 sec one-way constant delay in one direction of transmission.
3. measure (average) one-way delay with 2 or more implementations, using identical waiting time thresholds (Thresh) for loss set at 3 seconds.
4. configure the network emulator to add 3 sec one-way constant delay in one direction of transmission equivalent to 2 seconds of additional one-way delay (or change the path delay while test is in progress, when there are sufficient packets at the first delay setting)
5. repeat/continue measurements
6. observe that the increase measured in step 5 caused all packets with 2 sec additional delay to be declared lost, and that all packets that arrive successfully in step 3 are assigned a valid one-way delay.

The common parameters used for tests in this section are:

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

The netem emulator was set to add constant delays as specified in the procedure above.

6.2.1. NetProbe results for Loss Threshold

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

6.2.2. Perfas Results for Loss Threshold

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

6.2.3. Conclusions for Loss Threshold

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

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

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

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

The tests described in this section used:

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


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

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

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

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

Null Hypothesis: All samples come from a common population.

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

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

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

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

6.4. Poisson Sending Process Evaluation

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

In this section, each implementation's Poisson distribution is compared with an idealistic version of the distribution available in the base functionality of the R-tool for Statistical Analysis[Rtool],

and performed using the Anderson-Darling Goodness-of-Fit test package (ADGofTest) [Radgof]. The Goodness-of-Fit criterion derived from [RFC2330] requires a test statistic value $AD \leq 2.492$ for 5% significance. The Appendix of [RFC2330] also notes that there may be difficulty satisfying the ADGofTest when the sample includes many packets (when 8192 were used, the test always failed, but smaller sets of the stream passed).

Both implementations were configured to produce Poisson distributions with $\lambda = 1$ packet per second.

6.4.1. NetProbe Results

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

The statistical summary for two NetProbe streams is below:

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

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

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

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

```
Anderson-Darling GoF Test
```

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

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

```
Anderson-Darling GoF Test
```

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

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

```
Anderson-Darling GoF Test
```

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

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

```
Anderson-Darling GoF Test
```

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

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

6.4.2. Perfas Results

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

The statistical summary for two Perfas+ streams is below:

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

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

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

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

Anderson-Darling GoF Test

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

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

Anderson-Darling GoF Test

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

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

Anderson-Darling GoF Test

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

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

Anderson-Darling GoF Test

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

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

Anderson-Darling GoF Test

```
data: a27pe$p2[1:100] and pexp
```

```
AD = 0.4758, p-value = 0.7712
alternative hypothesis: NA
```

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

```
Anderson-Darling GoF Test
```

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

```
>
```

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

6.4.3. Conclusions for Goodness-of-Fit

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

6.5. Implementation of Statistics for One-way Delay

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

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

Implementation of Section 4 Statistics

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

7. Conclusions for RFC 2680bis

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

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

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

8. Security Considerations

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

9. IANA Considerations

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

10. Acknowledgements

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

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

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

11. References

11.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, March 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

- [RFC6808] Ciavattone, L., Geib, R., Morton, A., and M. Wieser, "Test Plan and Results Supporting Advancement of RFC 2679 on the Standards Track", RFC 6808, December 2012.

11.2. Informative References

- [ADK] Scholz, F. and M. Stephens, "K-sample Anderson-Darling Tests of fit, for continuous and discrete cases", University of Washington, Technical Report No. 81, May 1986.
- [I-D.morton-ippm-advance-metrics] Morton, A., "Lab Test Results for Advancing Metrics on the Standards Track", draft-morton-ippm-advance-metrics-02 (work in progress), October 2010.
- [Perfas] Heidemann, C., "Qualitaet in IP-Netzen Messverfahren", published by ITG Fachgruppe, 2nd meeting 5.2.3 (NGN) http://www.itg523.de/oeffentlich/01nov/Heidemann_QOS_Messverfahren.pdf , November 2001.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [Radgof] Bellosta, C., "ADGofTest: Anderson-Darling Goodness-of-Fit Test. R package version 0.3.", <http://cran.r-project.org/web/packages/ADGofTest/index.html>, December 2011.
- [Radk] Scholz, F., "adk: Anderson-Darling K-Sample Test and Combinations of Such Tests. R package version 1.0.", , 2008.
- [Rtool] R Development Core Team, "R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>", , 2011.
- [WIPM] "AT&T Global IP Network", <http://ipnetwork.bgtmo.ip.att.net/pws/index.html>, 2012.

Authors' Addresses

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

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

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

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

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

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

Matthias Wieser
Technical University Darmstadt
Darmstadt,
Germany

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

Network Working Group
Internet Draft
Intended status: Informational
Expires: November 2013

Tal Mizrahi
Marvell
May 28, 2013

UDP Checksum Trailer in OWAMP and TWAMP
draft-mizrahi-owamp-twamp-checksum-trailer-00.txt

Abstract

The One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP) are used for performance monitoring in IP networks. Delay measurement is performed in these protocols by using timestamped test packets. Some implementations use hardware-based timestamping engines that integrate the accurate transmission timestamp into every outgoing OWAMP/TWAMP test packet during transmission. Since these packets are transported over UDP, the UDP checksum field is then updated to reflect this modification. This document proposes to use the last 2 octets of every test packet as a Checksum Trailer, allowing timestamping engines to reflect the checksum modification in the last 2 octets rather than in the UDP checksum field. The behavior defined in this document is completely interoperable with existing OWAMP/TWAMP implementations.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

This Internet-Draft will expire on October 28, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	4
2.1. Terminology	4
2.2. Abbreviations	4
3. Using the UDP Checksum Trailer in OWAMP and TWAMP	5
3.1. Overview	5
3.2. OWAMP / TWAMP Test Packets with Checksum Trailer	5
3.2.1. Transmission of OWAMP/TWAMP with Checksum Trailer ..	8
3.2.2. Intermediate Updates of OWAMP/TWAMP with Checksum Trailer	8
3.2.3. Reception of OWAMP/TWAMP with Checksum Trailer	8
3.3. Interoperability with Existing Implementations.....	8
3.4. Using the Checksum Trailer with or without Authentication	8
4. Security Considerations	9
5. IANA Considerations	9
6. Acknowledgments	9
7. References	9
7.1. Normative References	9
7.2. Informative References	10

1. Introduction

The One-Way Active Measurement Protocol ([OWAMP]) and the Two-Way Active Measurement Protocol ([TWAMP]) are used for performance monitoring in IP networks.

Delay and delay variation are two of the metrics that OWAMP/TWAMP can measure. This measurement is performed using timestamped test packets.

The accuracy of delay measurements relies on the timestamping method and its implementation. In order to facilitate accurate timestamping, an implementation MAY use a hardware based timestamping engine, as shown in Figure 1. In such cases, the OWAMP/TWAMP packets are sent and received by a software layer, whereas the timestamping engine modifies every outgoing test packet by incorporating its accurate transmission time into the <Timestamp> field in the packet.

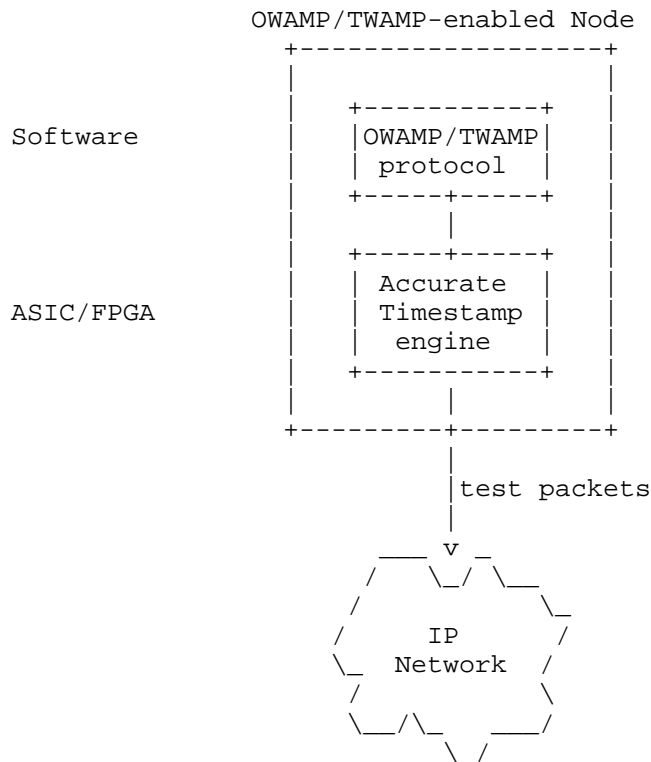


Figure 1 Accurate Timestamping in OWAMP/TWAMP

OWAMP/TWAMP test packets are transported over UDP. When the UDP payload is changed by an intermediate entity such as the timestamping engine, the UDP Checksum field must be updated to reflect the new payload. When using UDP over IPv4 ([UDP]), an intermediate entity that cannot update the value of the UDP checksum can assign a value of zero to the checksum field, causing the receiver to ignore the

checksum field. UDP over IPv6, as defined in [IPv6], does not allow a zero checksum, and requires the UDP checksum field to contain a correct checksum of the UDP payload.

Since an intermediate entity only modifies a specific field in the packet, i.e. the timestamp field, the UDP checksum update can be performed incrementally, using the concepts presented in [Checksum].

A similar problem is addressed in Annex E of [IEEE1588]. When the Precision Time Protocol (PTP) is transported over IPv6, two octets are appended to the end of the PTP payload for UDP checksum updates. The value of these two octets can be updated by an intermediate entity, causing the value of the UDP checksum field to remain correct.

This document defines a similar concept for [OWAMP] and [TWAMP], allowing intermediate entities to update OWAMP/TWAMP test packets and maintain the correctness of the UDP checksum by modifying the last 2 octets of the packet.

The term Checksum Trailer is used throughout this document and refers to the 2 octets at the end of the UDP payload, used for updating the UDP checksum by intermediate entities.

The usage of the Checksum Trailer can in some cases simplify the implementation, since if the packet data is processed in a serial order, it is simpler to first update the timestamp field, and then update the Checksum Trailer rather than to update the timestamp and then update the UDP checksum, residing at the UDP header.

2. Conventions used in this document

2.1. Terminology

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

2.2. Abbreviations

NTP	Network Time Protocol
OWAMP	One-Way Active Measurement Protocol
PTP	Precision Time Protocol
TWAMP	Two-Way Active Measurement Protocol

UDP User Datagram Protocol

3. Using the UDP Checksum Trailer in OWAMP and TWAMP

3.1. Overview

The UDP Checksum Trailer is a two-octet trailer that is piggybacked at the end of the test packet. It resides in the last 2 octets of the UDP payload.

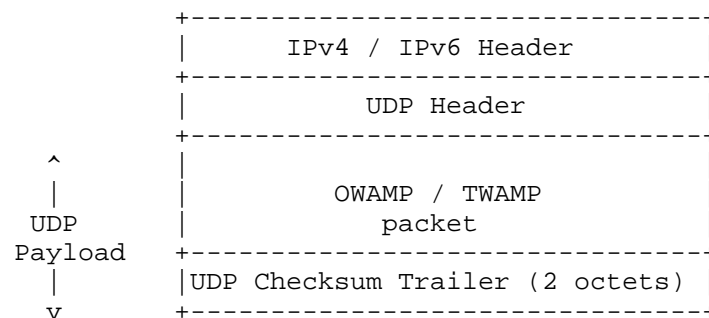


Figure 2 Checksum Trailer in OWAMP/TWAMP Test Packet

3.2. OWAMP / TWAMP Test Packets with Checksum Trailer

The One-Way Active Measurement Protocol [OWAMP], and the Two-Way Active Measurement Protocol [TWAMP] both make use of timestamped test packets. The formats of these packets are defined in [OWAMP] and in [TWAMP].

OWAMP/TWAMP test packets are transported over UDP, either over IPv4 or over IPv6. This document applies to both OWAMP/TWAMP over IPv4 and over IPv6.

OWAMP/TWAMP test packets contain a Packet Padding field. This document proposes to use the last 2 octets of the Packet Padding field as the Checksum Trailer. In this case the Checksum Trailer is always the last 2 octets of the UDP payload, and thus the trailer is located $\text{UDP Length} - 2$ octets after the beginning of the UDP header.

Figure 3 illustrates the OWAMP test packet format including the UDP checksum trailer.

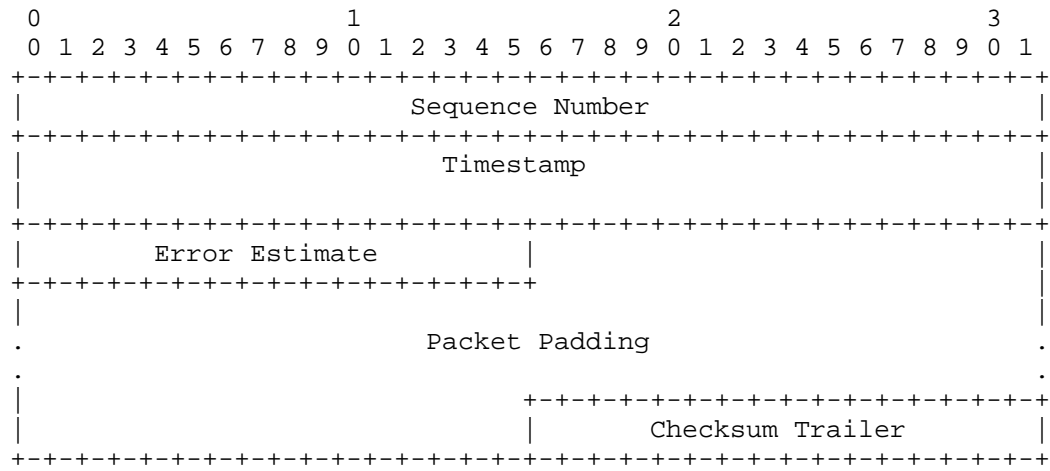


Figure 3 Checksum Trailer in OWAMP Test Packets

Figure 4 illustrates the TWAMP test packet format including the UDP checksum trailer.

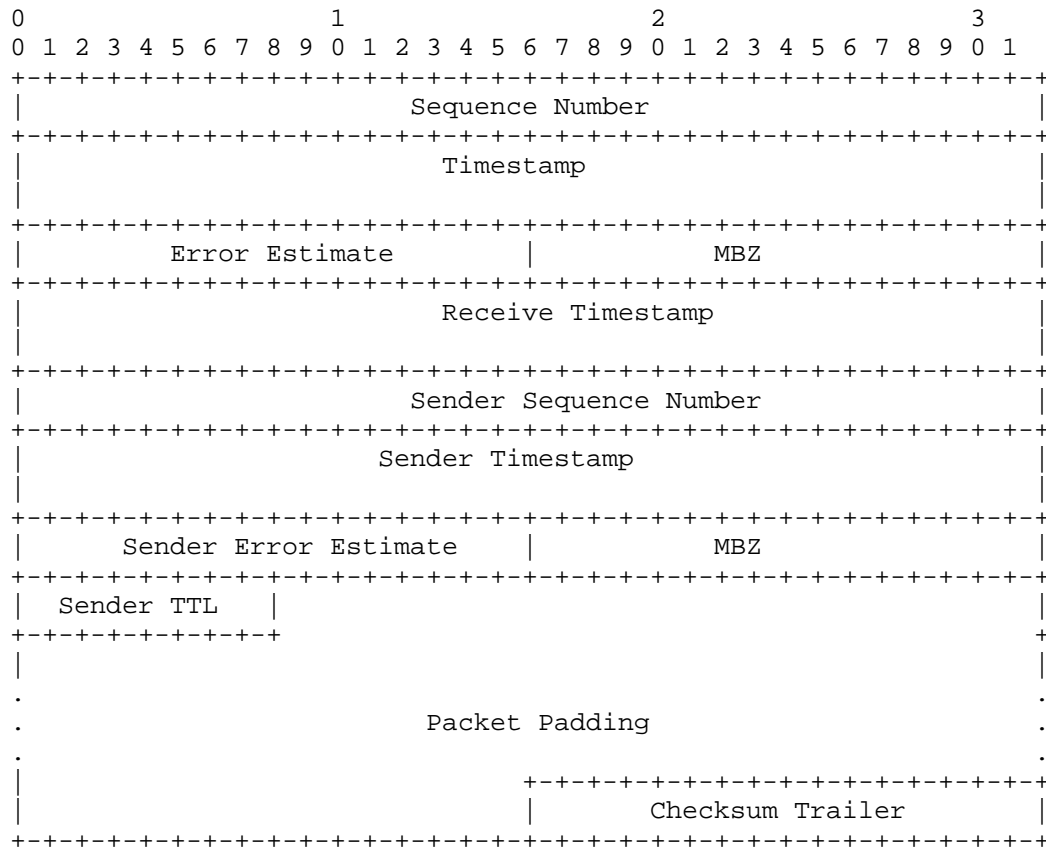


Figure 4 Checksum Trailer in TWAMP Test Packets

The length of the Packet Padding field in test packets is announced during the session initiation through the "Padding Length" field in the Request-Session message [OWAMP], or in the Request-TW-Session [TWAMP].

When a Checksum Trailer is included, the "Padding Length" MUST include the Checksum Trailer.

3.2.1. Transmission of OWAMP/TWAMP with Checksum Trailer

The transmitter of an OWAMP/TWAMP test packet MAY include a Checksum Trailer field, incorporated in the last 2 octets of the Packet Padding.

A transmitter that includes a Checksum Trailer in its outgoing test packets MUST include a Packet Padding in these packets, the length of which is at least 2 octets.

3.2.2. Intermediate Updates of OWAMP/TWAMP with Checksum Trailer

An intermediate entity that receives and alters an OWAMP/TWAMP test packet MAY alter the Checksum Trailer field in order to maintain the correctness of the UDP checksum value.

3.2.3. Reception of OWAMP/TWAMP with Checksum Trailer

This document does not impose new requirements on the receiving end of an OWAMP/TWAMP test packet.

The UDP layer at the receiving end verifies the UDP Checksum of received test packets, and the OWAMP/TWAMP layer SHOULD treat the Checksum Trailer as part of the Packet Padding.

3.3. Interoperability with Existing Implementations

The behavior defined in this document does not impose new requirements on the reception behavior of an OWAMP receiver or a TWAMP reflector, since the existence of the checksum trailer is transparent from the perspective of the receiver/reflector. Thus, the functionality described in this document allows interoperability with existing implementations that comply to [OWAMP] or [TWAMP].

3.4. Using the Checksum Trailer with or without Authentication

When message authentication is used, intermediate entities that alter test packets must also re-compute the Message Authentication Code (MAC) accordingly. The MAC update typically requires the intermediate entity to store the packet, re-compute its MAC, and then forward it.

While a Checksum Trailer MAY be used when authentication is enabled, in practice the Checksum Trailer is more useful in unauthenticated mode, allowing the intermediate entity to perform serial processing of the packet without storing-and-forwarding it.

4. Security Considerations

This document describes how the last two octets of a test packet can be used for updating the checksum. This concept is logically similar to an intermediate node that directly modifies the UDP Checksum field, and thus does not present any new security implications.

As described in Section 3.4. , the concept described in this document is especially useful for unauthenticated mode. However, this document does not make a statement about the circumstances in which authentication should or should not be used.

5. IANA Considerations

There are no IANA actions required by this document.

RFC Editor: please delete this section before publication.

6. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

7. References

7.1. Normative References

- | | |
|------------|--|
| [KEYWORDS] | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. |
| [IPv6] | Deering, S., Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998. |
| [Checksum] | Rijsinghani, A., "Computation of the Internet Checksum via Incremental Update", RFC 1624, May 1994. |
| [UDP] | Postel, J., "User Datagram Protocol", RFC 768, August 1980. |
| [OWAMP] | Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and Zekauskas, M., "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006. |
| [TWAMP] | Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and Babiarz, J., "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008. |

7.2. Informative References

- [IEEE1588] IEEE TC 9 Instrumentation and Measurement Society,
"1588 IEEE Standard for a Precision Clock
Synchronization Protocol for Networked Measurement and
Control Systems Version 2", IEEE Standard, 2008.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

