Network Working Group
INTERNET-DRAFT
Intended Status: Standards Track
Expires: September 5, 2011

S. Baillargeon
C. Flinta
A. Johnsson
S. Ekelin
Ericsson

March 4, 2011

TWAMP Value-Added Octets
draft-baillargeon-ippm-twamp-value-added-octets-01.txt

Abstract

This memo describes the optional extensions to the standard TWAMP test protocol for identifying test sessions and packet trains, and for measuring capacity metrics like the available path capacity, tight section capacity and UDP throughput in the forward and reverse path directions.

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/1id-abstracts.html

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

Table of Contents

1 Introduction

The notion of embedding a number of meaningful fields in the padding octets has been established as a viable methodology for carrying additional information within the TWAMP-Test protocol running between a Session-Sender and a Session-Reflector [RFC5357] [RFC6038].

This memo describes an OPTIONAL feature for the Two-Way Active Measurement Protocol [RFC5357]. It is called the Value-Added Octets feature.

This feature enables the controller host to measure capacity metrics like the IP-type-P available path capacity (APC) [RFC5136], IP-layer tight section capacity (TSC) [Y1540] and UDP throughput [RFC1242] on both forward and reverse paths. With this feature, it is also possible to improve the demultiplexing of test packets to the correct test sessions running on the controller and responder hosts when methods solely based on IP and UDP header information is not desirable or insufficient.

The Valued-Added Octets feature consists of new behaviors for the Session-Sender and Session-Reflector, and a set of value-added octets of information that are placed at the beginning of the Packet Padding field [RFC5357] or at the beginning of the Packet Padding (to be reflected) field [RFC6038] by the Session-Sender, and are reflected or returned by the Session-Reflector. The length of the value-added octets varies in size between 6, 10 and 14 octets depending on the setting of the flag bits specified at the beginning of the value-added octets.

This memo is an update to the TWAMP core protocol specified in [RFC5357]. Measurement systems are not required to implement the feature described in this memo to claim compliance with [RFC5357].

UDP throughput is defined in the Benchmarking Terminology for Network Interconnection Devices [RFC1242]. IP-Type-P APC metric is defined in Defining Network Capacity [RFC5136]. IP-layer TSC metric is defined in IP Packet Transfer and Availability Performance Parameters [Y1540]. The actual method to calculate the available path capacity, the tight section capacity or the UDP throughput from packet-level data performance data is not discussed in this memo.

1.1 Requirements Language

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

2 Purpose and scope

The purpose of this memo is to define the OPTIONAL Valued-Added Octets feature for TWAMP [RFC5357].

The scope of the memo is limited to specifications of the following enhancements:

- The extension of the modes of operation through assignment of a new value in the Mode field to communicate feature capability and use,
- The definition of a structure for embedding a sequence of value-added fields at the beginning of the Packet Padding field [RFC5037] or Packet Padding (to be reflected) field [RFC6038] in the TWAMP test packets and,
- The definition of new Session-Sender and Session-Reflector behaviors

The motivation for this feature is to enable the measurements of capacity metrics on both the forward and reverse paths, and to improve the demultiplexing of test packets to the correct test session at both endpoints.

This memo extends the modes of operation through assignment one new value in the Modes field (see Section 3.1 of [RFC4656] for the format of the Server Greeting message), while retaining backward compatibility with the core TWAMP [RFC5357] implementations. The new value correspond to the Valued-Added Octets Version 1 feature defined in this memo.

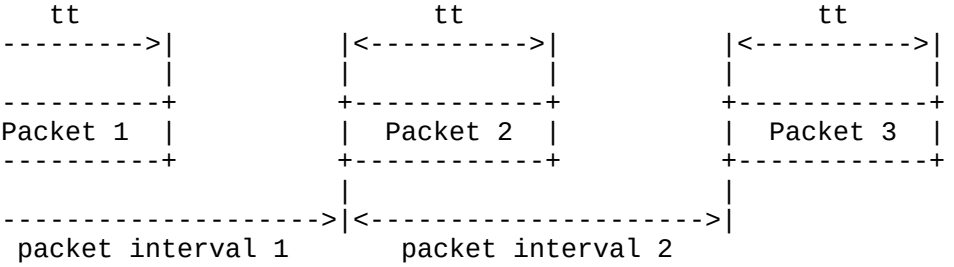
When the Server and Control-Client have agreed to use the Value-Added Octets Version 1 mode during control connection setup, then the Control-Client, the Server, the Session-Sender, and the Session-Reflector MUST all conform to the requirements of that mode, as identified below.

The OPTIONAL packet padding octets are designed to retain backward compatibility with the original TWAMP test protocol [RFC5357].

3 Capacity Measurement Principles

Most capacity estimation methods for available path capacity [RRBNC][PDM][ENHJMMB][SBW] and for UDP throughput [RFC2544] need to send and receive packets in groups, called packet trains or simply trains. Each train is sent at a specific transmission rate in a given direction. These trains must be identified within each bi-directional test session stream.

The first measurement principle is to send multiple trains within a test session stream from one IP node to another IP node in order to estimate the available path capacity, tight section capacity or UDP throughput in the forward direction. Each train consists of a group of test packets which are separated from each other by a packet interval, as shown in the picture below.



The packet interval between consecutive packets for each train sent by the Session-Sender and reflected by the Session-Reflector MUST be calculated and determined by the controller or an application or entity communicating with the controller. The packet interval MAY be constant within a train. Determination of the packet interval within a train as well as for consecutive trains for a given test session is implementation-specific.

The transmission time tt to send one packet (i.e. determined by the interface speed and the IP packet size) is also shown in the picture. Observe that the packet interval MUST be larger than or equal to tt.

At the Session-Reflector, each received test packet within a forward train is time stamped. This provides a second set of packet interval values. Methods for measuring the available path capacity, tight section capacity and UDP throughput use the packet intervals obtained from both end points in the estimation process. The method to measuring the UDP throughput may also require the packet loss at the receiving end. The estimation process itself as well as any requirements on software or hardware is implementation-specific.

The second measurement principle is referred to as self-induced congestion. According to this principle, in order to measure the available path capacity, tight section capacity and UDP throughput, some trains MUST cause momentary congestion on the network path. In essence this means that some trains MUST be sent at a higher rate than what is available on the network path. The congestion is only transient, for the duration of the train which is typically short.

In order to fulfill the above measurement principles and to measure the available path capacity, tight section capacity and UDP throughput in the reverse direction, the reflected test packets MUST be re-grouped into trains at the Session-Reflector.

4 Test packet Demultiplexing Principles

The controller (or the Session-Sender) requires a method for demultiplexing the received test packets to the correct test session especially when it manages multiple active test sessions. The responder also requires a method for demultiplexing the received test packets from multiple active test sessions originating from the same controller or from different controllers.

The purpose of this section is to provide some basic principles for identifying the test packets and to clarify the optional usage of the Sender Discriminator (SD) field described in this memo. It is important to note the actual method for identifying a test packet and the process for mapping it to the correct test session are implementation-specific. They may differ between various controllers and responders.

In general, the methods are based on fields available in the various headers of the TWAMP test packet (e.g. Ethernet, IP, UDP and TWAMP headers). Note the SID [RFC4656] is generally not used for identification purpose since it does not normally appear in the TWAMP test packets. As an example, a measurement system (controller or responder) may use the source IP address of the incoming test packet in order to associate it to the correct test session. This method is valid but has a number of limitations. It is simple and effective when each measurement system only requires a single test session for each peer but fails when multiple test sessions (with different characteristics) are running between the same pair of controller and responder.

Another approach is to use a combination of the source IP address, destination IP address, source UDP port and destination UDP port. This method is also valid but to work effectively, it requires that the controller allocates multiple UDP ports (one for each test session for instance) and/or the responder listens on multiple ports.

Ideally, a measurement system should limit the number of UDP ports for sending and receiving test packets. This approach may be improved by using a combination of the IP addresses, UDP ports and DSCP codepoint. This method also has its limitations. For instance, it cannot identify test packets from different test sessions running between the same pair of controller and responder if they are using the same UDP endpoints and the same DSCP codepoint.

This memo introduces a new field, the Sender Discriminator (SD) field intended to simplify the identification of the test packets at the controller and responder. It is especially useful when multiple test sessions with different DSCP codepoints and/or test packet sizes are expected to be running between the same pair of UDP endpoints. As described in 6.1.2, the SD is a number generated by the Session-Sender that uniquely identifies a test session on its system. With this field, the controller can explicitly identify the test packets belonging to a test session. When provided, the responder MAY use the SD field in combination of the source IP address for instance to identify the test packets belonging to a test session.

5 TWAMP Control Extensions

TWAMP-Control protocol [RFC5357] uses the Modes field to identify and select specific communication capabilities, and this field is a recognized extension mechanism.

TWAMP connection establishment follows the procedure defined in Section 3.1 of [RFC4656] and Section 3.1 of [RFC5357]. The new feature require one new bit position (and value) to identify the ability of the Server/Session-Reflector to read and act upon the new fields in the value-added octets. See the IANA section for details on the assigned value and bit position.

The Server sets the new bit position in the Modes field of the Server Greeting message to indicate its capability to operate in this new mode.

Both the Reflect Octets mode and Symmetrical Size mode SHOULD be selected to ensure the reflection of the value-added padding octets by the Session-Reflector and symmetrical size TWAMP-Test packets in the forward and reverse directions of transmission.

The forward and reverse APC, TSC and UDP throughput measurement characteristics depend on the size of the test packets. All test packets (forward and reverse test packets) belonging to a specific test session responsible to measure the available path capacity, tight section capacity and/or UDP throughput MUST have the same IP

packet size.

6 Extended TWAMP Test

The TWAMP-test protocol carrying the value-added padding octets is identical to TWAMP [RFC5357] except for the definition of first 6, 10 or 14 octets in the Padding Octet field that the Session-Sender expects to be reflected.

The Session-Sender and Session-Reflector behaviors are also modified.

6.1 Sender Behavior

This section describes the extensions to the behavior of the TWAMP Session-Sender.

When the Value-Added Octets Version 1 mode is selected, the Session-Sender MAY set the Sender Discriminator Present bit to 1. If it is set to 1, the Session-Sender MUST generate and transmit a unique nonzero discriminator value in the Sender Discriminator field.

When the Value-Added Octets Version 1 mode is selected, the Session-Sender MAY set the Last Seqno in Train Present bit to 1. If it is set to 1, the Session-Sender MUST generate and transmit a valid sequence number in the Last Seqno in Train field. The Session-Sender MUST also group the test packets in trains and send the trains towards the Session-Reflector at the desired forward packet intervals.

When the Value-Added Octets Version 1 mode is selected, the Session-Sender MAY set the the Desired Reverse Packet Interval Present bit to 1. If it is set to 1, the Session-Sender MUST generate and transmit a valid inter-packet time interval in the Desired Reverse Packet Interval field.

The desired forward and reverse rate interval parameters are usually provided by a measurement method, tool or algorithm. This measurement algorithm is outside the scope of this specification.

6.1.1 Packet Timings

The Send Schedule is not utilized in TWAMP and this is unchanged in this memo.

6.1.2 Session-Sender Packet Format

The Session-Sender packet format follows the same procedure and guidelines as defined in TWAMP [RFC5357] and TWAMP Reflect Octets and

Symmetrical Size Features [RFC6038].

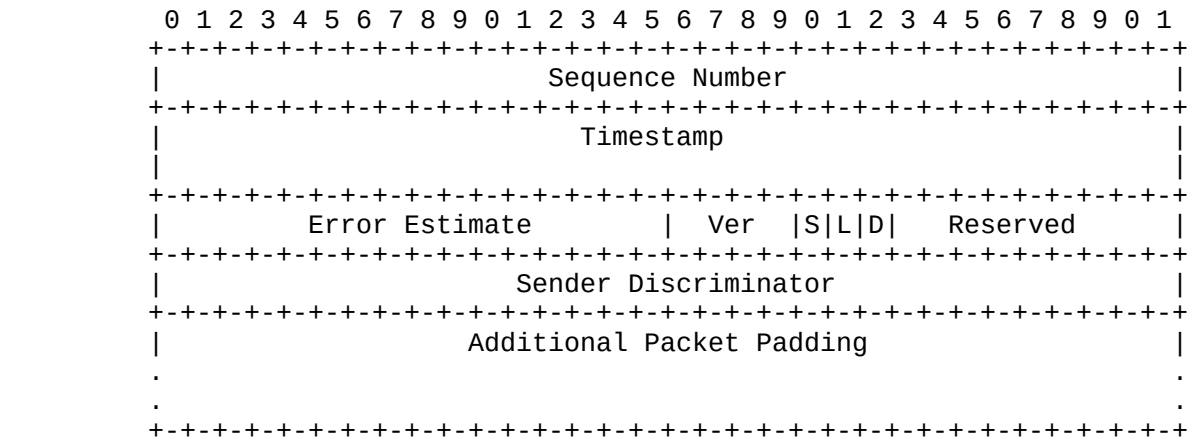
This feature allows the Session-Sender to set the first few octets in the TWAMP-Test Packet Padding field with information to communicate value-added padding version number, flag bits, sender discriminator, sequence number of the last packet in a train and desired inter-packet time interval (or per-packet waiting time) for the reverse path direction of transmission.

A version number and a sequence of flag bits are defined at the very beginning of the value-added padding octets. The version number identifies the version of the value-added padding octets and meaning of the flag bits. Each flag bit indicates if a specific field of a specific size is present in the valued-added padding octets. The flag bits are designed to accommodate different combinations of fields while reducing padding overhead when certain fields are not needed. The version number and flag bits also provide an effective method for parsing information at Session-Reflector and Session-Sender. This document defines version 1 with three flag bits: S, L and D.

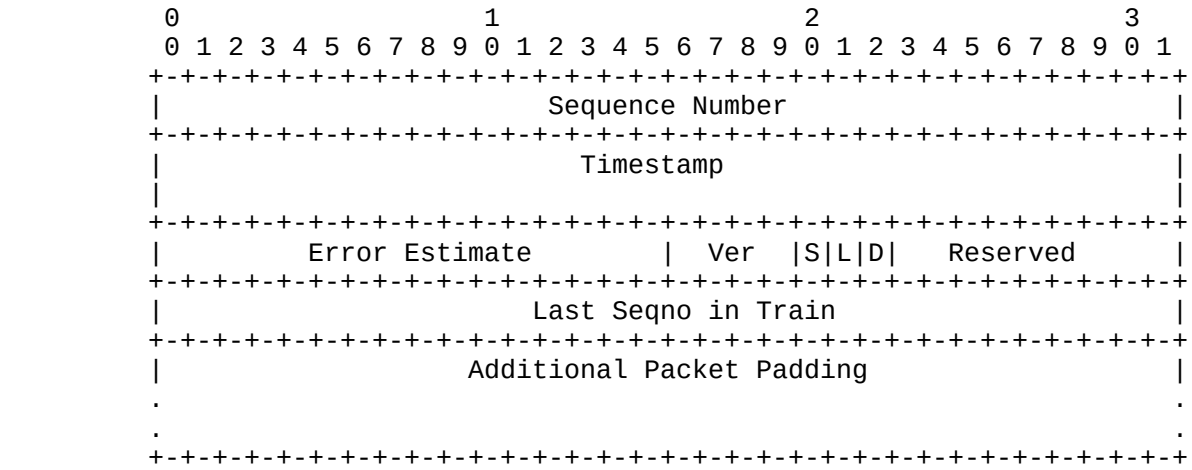
The format of the test packet depends on the TWAMP mode and flag bits being used. The Value-Added Octets Version 1 mode is intended to work with any TWAMP modes.

When the Value-Added Octets Version 1 is selected with S=1, L=0 and D=0, the Session-Sender SHALL use the following TWAMP test packet format in unauthenticated mode:

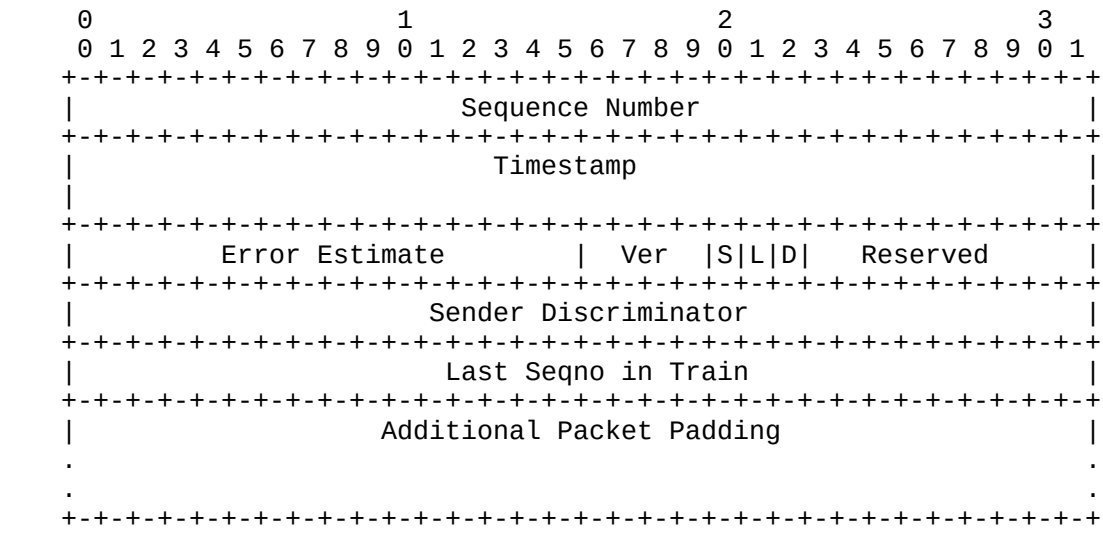
0 1 2 3



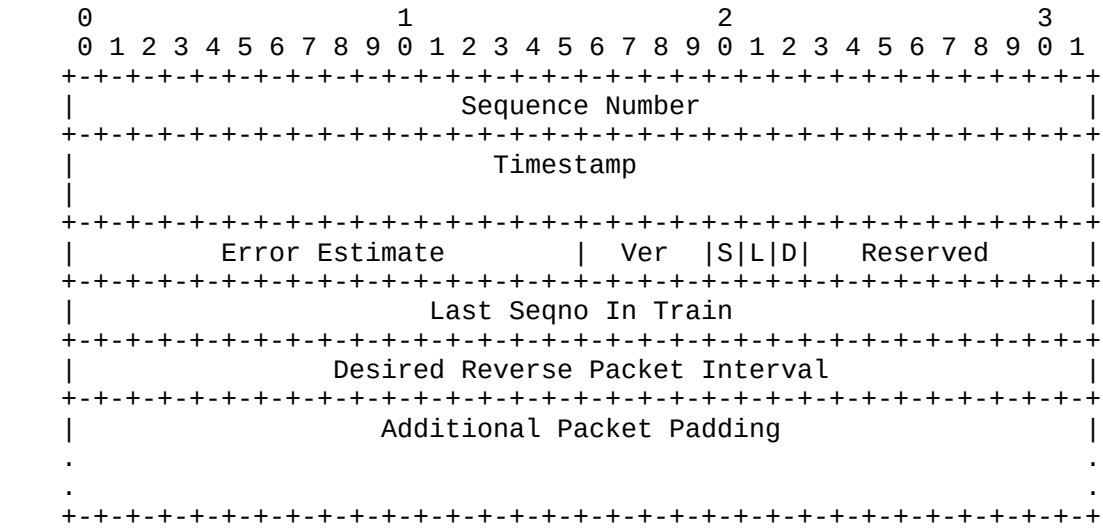
When the Value-Added Octets Version 1 is selected with S=0, L=1 and D=0, the Session-Sender SHALL use the following TWAMP test packet format in unauthenticated mode:



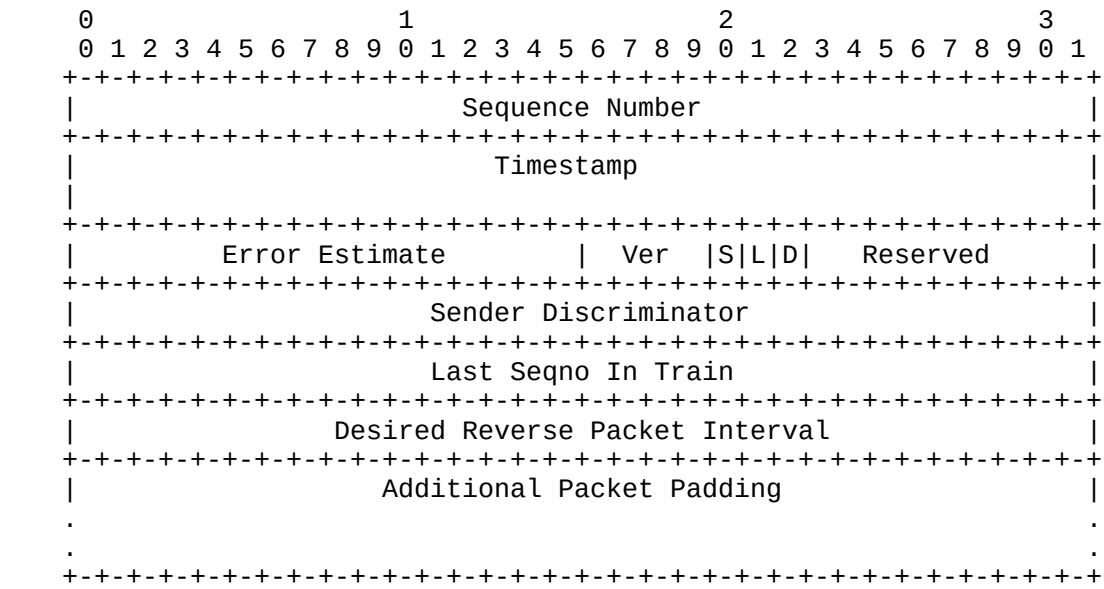
When the Value-Added Octets Version 1 is selected with S=1, L=1 and D=0, the Session-Sender SHALL use the following TWAMP test packet format in unauthenticated mode:



When the Value-Added Octets Version 1 is selected with S=0, L=1 and D=1, the Session-Sender SHALL use the following TWAMP test packet format in unauthenticated mode:

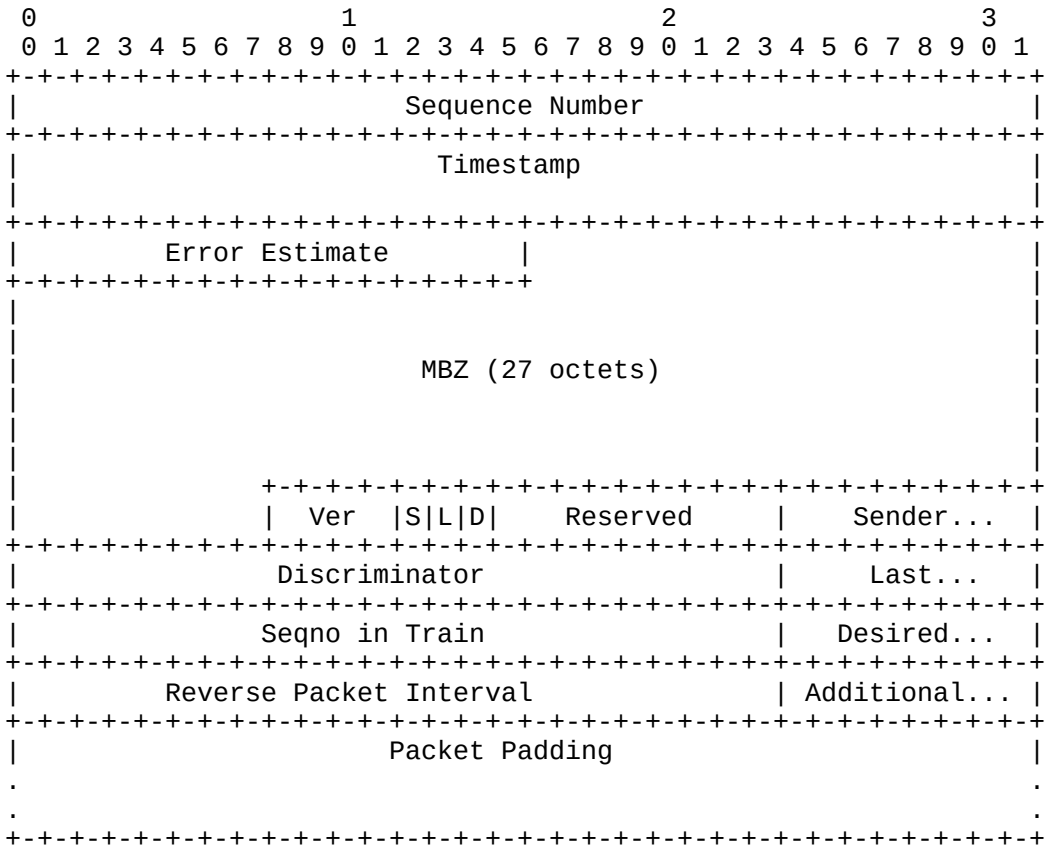


When the Value-Added Octets Version 1 is selected with S=1, L=1 and D=1, the Session-Sender SHALL use the following TWAMP test packet format in unauthenticated mode:



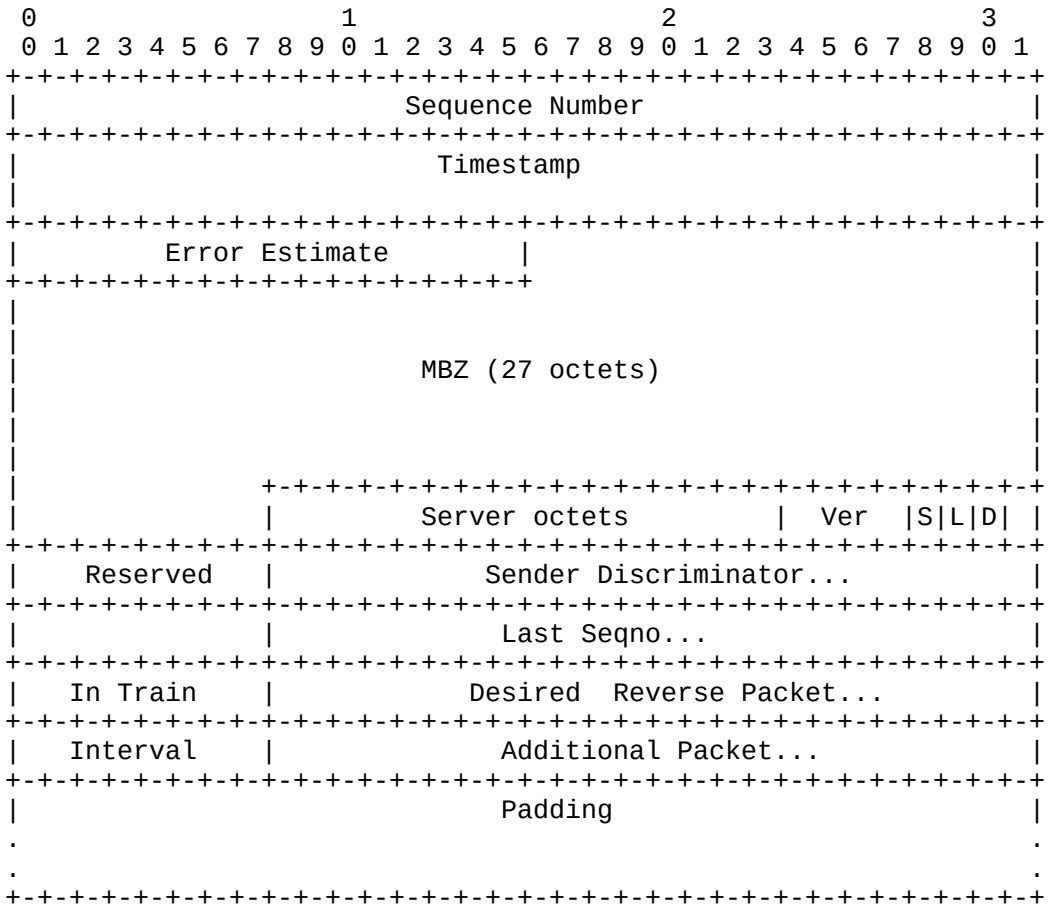
When the Value-Added Octets Version 1 is selected with S=1, L=1 and

D=1, the Session-Sender SHALL use the following TWAMP test packet format in conjunction with the unauthenticated mode, Symmetrical Size mode and Reflect Octets mode:



When the Value-Added Octets Version 1 is selected with S=1, L=1 and

D=1, the Session-Sender SHALL use the following TWAMP test packet format in conjunction with the unauthenticated mode, Symmetrical Size mode and Reflect Octets mode with a non-zero value in the Server octets field:



In the combined mode including Reflect Octets, the value-added padding octets are embedded in the Packet Padding (to be reflected) field.

The Version (Ver) field MUST be encoded in the first 4 bits. It identifies the version number of the value-added padding octets and meaning of the flag bits. This document defines version 1.

The Sender Discriminator Present bit (S) MUST be the first flag. If the Sender Discriminator Present bit is set to 1, then a Sender Discriminator field MUST be present and MUST contain valid information.

The Last Seqno in Train Present bit (L) MUST be the second flag. If the Last Seqno in Train Present bit is set to 1, then the Last Seqno in Train field MUST be present and MUST contain valid information.

The Desired Reverse Packet Interval Present bit (D) MUST be the third flag. If the Desired Reverse Packet Interval Present bit is set to 1, then Desired Reverse Packet Interval Present field MUST be present and MUST contain valid information.

The Reserved field is reserved for future use. All 9 bits of the Reserved field MUST be transmitted as zero by the Session-Sender.

The Sender Discriminator (SD) field MUST contain an unsigned 32 bit integer generated by the Session-Sender. It is used by the Session-Reflector and/or Session-Sender to identify packets belonging to a test session. The Session-Sender MUST choose a nonzero discriminator value that is unique among all test sessions on its system. This field is present only if the Sender Discriminator Present bit is set to one.

The Last Seqno in Train MUST contain an unsigned 32 bit integer generated by the Session-Sender. It MUST indicate the expected sequence number of the last packet in the train. It SHOULD be used by the Session-Sender and Session-reflector to identify the train a test packet belongs to. The packets belonging to a train are determined by observing the test packet sequence number in relation to the Last Seqno for a train. The sequence number of a packet in a train MUST be lower than or equal to the Last Seqno for that train. The sequence number MUST also be larger than the Last Seqno for the previous train. This field is present only if the Last Seqno in Train Present bit is set to one.

The Desired Reverse Packet Interval (DRPI) MUST contain an unsigned 32 bit integer generated by the Session-Sender. It MUST indicate the desired inter-packet time interval (or the waiting time) that the Session-Reflector SHOULD use when transmitting the reflected test packets towards the Session-Sender. The value 0 means the The Session-Reflector SHOULD return the test packet to the Session-Sender as quickly as possible. The format of this field MUST be a fractional

part of a second as defined in OWAMP [RFC4656]. This field is present only if the Desired Reverse Packet Interval Present bit is set to one.

The method by which the Sender Discriminator and Desired Reverse Packet Interval values are obtained is outside of the scope of this document.

6.2 Reflector behavior

The TWAMP Session-Reflector follows the procedures and guidelines in Section 4.2 of [RFC5357], with some changes and additional functions.

When the Value-Added Octets Version 1 is selected, the behavior of the Session-Reflector SHALL be as follows:

- The Session-Reflector MUST read the Version field. If Ver=1, the Session-Reflector MUST read the S, L and D flag bits. If Ver is not equal 1, the Session-Reflector MUST ignore the rest of the value-added padding octets and MUST follow the procedures and guidelines described in section 4.2 of [RFC5357]. The Session-Reflector SHOULD transmit the packet as quickly as possible including the test packets that are currently stored for the test session.

- If S=0, L=0 and D=0, the Session-Reflector MUST ignore the rest of the value-added padding octets and MUST follow the procedures and guidelines described in section 4.2 of [RFC5357]. The Session-Reflector SHOULD transmit the packet as quickly as possible including the test packets that are currently stored for the test session.

- If S=1, the Session-Reflector MUST continue reading and extracting the information from the Sender Discriminator field in the value-added padding octets.

- After reading and extracting the information from the Sender Discriminator field, the Session-Reflector SHOULD associate the test packets to the correct test session based on the value specified in the Sender Discriminator field and the source IP address specified in the IP header of the test packet. The actual method for demultiplexing the received test packets to the correct test session based on the Sender Discriminator and source IP address is outside the scope of this specification. The Session-Reflector MAY also use additional packet fields to demultiplex test packets to a test session.

- If L=1, the Session-Reflector MUST continue reading and extracting the information from the Last Seqno in Train field in the value-added padding octets.

- After reading and extracting the information from the Last Seqno in Train field, Last Seqno in Train field MUST be compared to Sequence number in the same packet in order to determine when a complete train has been collected. The Session-Reflector SHOULD buffer the packets belonging to the current train (or store the packet-level performance data) and SHOULD transmit them as immediately as possible after the last packet of the train has been received. The last packet within a train has Sender Sequence Number = Last Seqno in Train.

- The Last Seqno in Train of a packet MUST also be compared to the Last Seqno in Train of the previous packet in order to determine if a new train needs to be collected. In case of packet loss, the Session-Reflector MUST transmit the incomplete train when it receives a packet with a Last SeqNo in Train belonging to the another train (e.g. next train) of the test session, or after a timeout. The timeout MAY be the REFWAIT timer specified in section 4.2 of [RFC5357].

- Packets arriving out-of-order within a train MUST be buffered at the Session-Reflector if the train is not yet transmitted to the Session-Sender. If the train is already transmitted, the test packet SHOULD be returned to the Session-Sender as quickly as possible. The Session-Reflector MUST not reorder the test packets if they happen to arrive out-of-sequence.

- Duplicate packets within a train MUST be buffered at the Session-Reflector if the train is not yet transmitted to the Session-Sender. If the train is already transmitted, the duplicate test packet SHOULD be returned to the Session-Sender as quickly as possible. The Session-Reflector MUST not discard duplicate test packets.

- If D=1, the Session-Reflector MUST continue reading and extracting the information from the Desired Reverse Packet Interval field in the value-added padding octets.

- After reading and extracting the information from the Desired Reverse Packet Interval field, the Session-Reflector SHOULD transmit the packets belonging to a reverse train with a waiting time (packet interval) for each packet indicated in the Desired Reverse Packet Interval field. If the Desired Reverse Packet Interval field is set to zero, then the Session-Reflector SHOULD transmit the packets as quickly as possible.

The Session-Reflector MUST implement the changes described above when the Value-Added Octets Version 1 mode is selected.

6.2.1 Session-Reflector Packet Format

The Session-Reflector packet format follows the same procedure and guidelines as defined in TWAMP [RFC5357] and TWAMP Reflect Octets and Symmetrical Size Features [RFC6038], with the following changes:

- The Session-Reflector MUST re-use (reflect) the value-added padding octets (6, 10 or 14 octets) provided in the Sender's Packet Padding.
- The Session-Reflector MAY re-use the rest of the padding octets in the Sender's Packet Padding.

When using the recommended truncation process [RFC5357], the Session-Reflector MUST truncate exactly 27 octets of padding in Unauthenticated mode,and exactly 56 octets in Authenticated and Encrypted modes.

6.3 Additional Considerations

It is not required to use the Sender Discriminator field for calculating the capacity metrics. The Sender Discriminator Present bit can be set to zero. However, the Session-Sender and Session-Reflector MUST implement a local policy to identify the test packets belonging to a specific test session. The method for demultiplexing the received test packets to the correct test session based on other packet fields (e.g. fields in the IP header) is outside the scope of this specification.

Capacity measurements introduce an additional consideration when the test sessions operate in TWAMP Light. When the Session-Reflector does not have knowledge of the session state, the measurement system will only be capable to estimate or calculate the capacity metrics in the forward path direction of transmission. Capacity measurements in the reverse path direction requires the Session-Reflector to have knowledge of the session state and be capable to identify the test packets belonging to a specific test session. The method for creating a session state from the initial test packets on the TWAMP Light Session-Reflector is outside the scope of this specification.

7 Security Considerations

The value-added padding octets permit new attacks on the responder host communicating with core TWAMP [RFC5357]. The responder host MUST provide a mechanism to protect or limit the use of its local memory or buffer space.

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

8 IANA Considerations

This memo adds one mode to the IANA registry for the TWAMP Modes field, and describes behavior when the new modes are used. This field is a recognized extension mechanism for TWAMP.

8.1. Registry Specification

IANA has created a TWAMP-Modes registry (as requested in [RFC5618]). TWAMP-Modes are specified in TWAMP Server Greeting messages and Setup Response messages, as described in Section 3.1 of [RFC5357], consistent with Section 3.1 of [RFC4656], and extended by this memo. Modes are indicated by setting bits in the 32-bit Modes field that correspond to values in the Modes registry. For the TWAMP-Modes registry, we expect that new features will be assigned increasing registry values that correspond to single bit positions, unless there is a good reason to do otherwise (more complex encoding than single-bit positions may be used in the future to access the 2^32 value space).

8.2. Registry Contents

The TWAMP-Modes registry has been augmented as follows:

Value	Description	Semantics Definition
128	Valued-Added Octets Ver 1	This memo, Section 2 new bit position (7)

9 References

9.1 Normative References

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

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol(OWAMP)", RFC 4656, September 2006.

[RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, July 1991.

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

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

[RFC6038] Morton, A., Ciavattone, L., TWAMP Reflect Octets and Symmetrical Size Features, RFC6038 , October 2010.

[RFC2544] Bradner,S., McQuaid, J., "Benchmarking Terminology for Network Interconnect Devices", RFC 2544, March 1999.

9.2 Informative References

[RRBNC] Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrel, L., Pathchirp: Efficient available bandwidth estimation for network paths, Passive and Active Measurement Workshop, 2003.

[PDM] Dovrolis, C., Ramanathan, P., and Moore D., Packet Dispersion Techniques and a Capacity Estimation Methodology, IEEE/ACM Transactions on Networking, December 2004.

[ENHJMMB] Ekelin, S., Nilsson, M., Hartikainen, E., Johnsson, A., Mangs, J., Melander, B., Bjorkman, M., Real-time measurement of end-to-end available bandwidth using kalman filtering, Proceedings to the IEEE IFIP Network Operations and Management Symposium, 2006.

[SBW] Sommers, J., Barford, P., Willinger, W., Laboratory-based calibration of available bandwidth estimation tools, Microprocess Microsyst., 2007.

[Y1540] ITU-T Y.1540, Internet protocol data communication service - IP packet transfer and availability performance parameters, 2011.

[MRM] Morton, A., Ramachandran, G., Maguluri, G., Reporting

Metrics Different Points of View, draft-ietf-ippm-reporting-metrics-03, June 2010.

Author's Addresses

Steve Baillargeon
Ericsson
3500 Carling Avenue
Ottawa, Ontario K2H 8E9
Canada
EMail: steve.baillargeon@ericsson.com

Christofer Flinta
Ericsson
Farogatan 6
Stockholm, 164 80
Sweden
EMail: christofer.flinta@ericsson.com

Andreas Johnsson
Ericsson
Farogatan 6
Stockholm, 164 80
Sweden
EMail: andreas.a.johnsson@ericsson.com

Svante Ekelin
Ericsson
Farogatan 6
Stockholm, 164 80
Sweden
EMail: svante.ekelin@ericsson.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 31, 2011

L. Ciavattone
AT&T Labs
R. Geib
Deutsche Telekom
A. Morton
AT&T Labs
M. Wieser
University of Applied Sciences
Darmstadt

June 29, 2011

Test Plan and Results for Advancing RFC 2679 on the Standards Track
draft-morton-ippm-testplan-rfc2679-01

Abstract

This memo proposes to advance a performance metric RFC along the standards track, specifically RFC 2679 on One-way Delay 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 2679.

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 December 31, 2011.

Table of Contents

1. Introduction

The IETF (IP Performance Metrics working group, IPPM) has considered how to advance their metrics along the standards track since 2001, with the initial publication of Bradner/Paxson/Mankin's memo [ref to work in progress, draft-bradner-metricstest-]. The original proposal was to compare the results of implementations of the metrics, because the usual procedures for advancing protocols did not appear to apply. It was found to be difficult to achieve consensus on exactly how to compare implementations, since there were many legitimate sources of variation that would emerge in the results despite the best attempts to keep the network paths equal, and because considerable variation was allowed in the parameters (and therefore implementation) of each metric. Flexibility in metric definitions, essential for customization and broad appeal, made the comparison task quite difficult.

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 *preliminary* consensus [I-D.ietf-ippm-metrictest] 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 [RFC2679]. It was prepared to help progress discussions on the topic of metric advancement, both through e-mail and at the upcoming IPPM meeting at IETF.

In particular, consensus is sought 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.

1.1. RFC 2679 Coverage

This plan, in it's first draft version, does not cover all critical requirements and sections of [RFC2679]. Material will be added as it is "discovered" (not all requirements use requirements language).

2. A Definition-centric metric advancement process

The process described in Section 3.5 of [I-D.ietf-ippm-metrictest] 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,

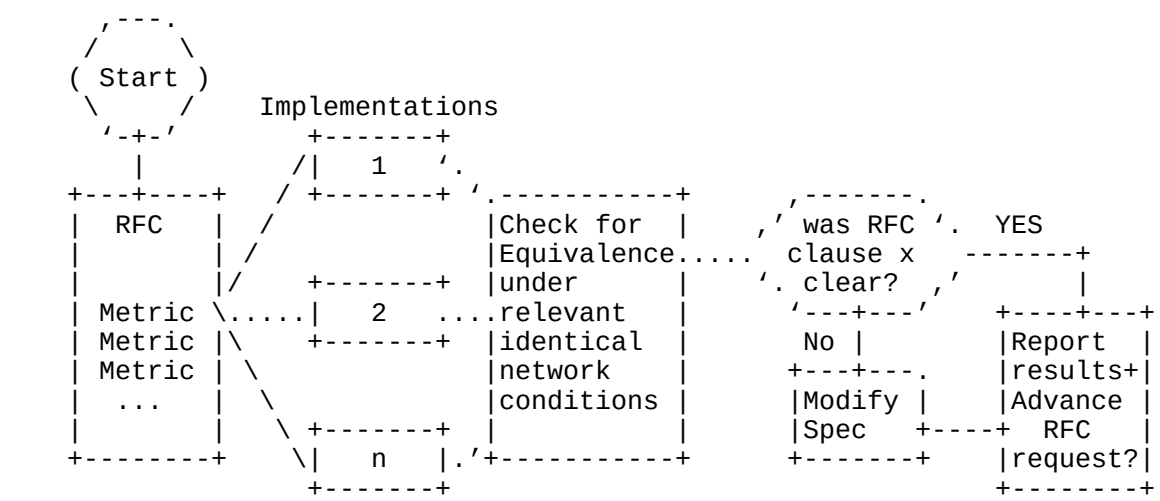
THEN the details of each implementation should be audited along with the exact definition text, to determine if there is a lack of clarity that has caused the implementations to vary in a way that affects the correspondence of the results.

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).

The figure below illustrates this process:

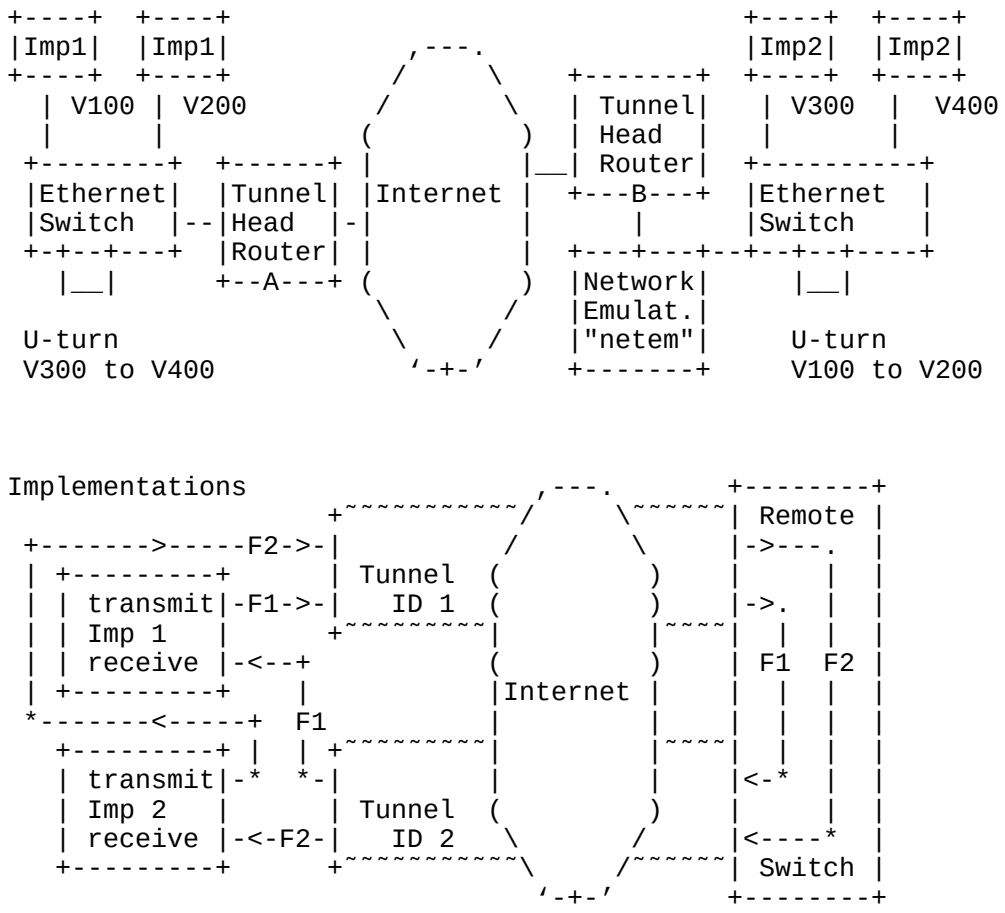


3. Test configuration

One metric implementation used was NetProbe version 5.8.5, (an earlier version is used in the WIPM system and deployed world-wide). 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+ 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 2 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 [I-D.ietf-ippm-metrictest].



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.

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-Delay-<StreamType>-Stream

With (Section 4.2. [RFC2679]) 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 3.82 of

[RFC2679]) And (Section 4.3. [RFC2679])

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

+ T, a time, and

+ dT, either a real number or an undefined number of seconds.

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

Also, Section 3.8.4 of [RFC2679] 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 with, and 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 2679

An implementation is required to report on its error calibration in Section 3.8 of [RFC2679] (also required in Section 4.8 for sample metrics). Sections 3.6, 3.7, and 3.8 of [RFC2679] give the detailed formulation of the errors and uncertainties for calibration. In summary, Section 3.7.1 of [RFC2679] describes the total time-varying uncertainty as:

Esynch(t)+ Rsource + Rdest

where:

Esynch(t) denotes an upper bound on the magnitude of clock synchronization uncertainty.

Rsource and Rdest denote the resolution of the source clock and the destination clock, respectively.

Further, Section 3.7.2 of [RFC2679] describes the total wire-time

uncertainty as

Hsource + Hdest

referring to the upper bounds on host-time to wire-time for source and destination, respectively.

Section 3.7.3 of [RFC2679] describes a test with small packets over an isolated minimal network where the results can be used to estimate systematic and random components of the sum of the above errors or uncertainties. In a test with hundreds of singletons, the median is the systematic error and when the median is subtracted from all singletons, the remaining variability is the random error.

The test context, or Type-P of the test packets, must also be reported, as required in Section 3.8 of [RFC2679] and all metrics defined there. Type-P is defined in Section 13 of [RFC2330] (as are many terms used below).

4.1. NetProbe Error and Type-P

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.

In general, NetProbe error is dependent on the specific version and installation details.

NetProbe operates using host time above the UDP layer, which is different from the wire-time preferred in [RFC2330], but can be identified as a source of error according to Section 3.7.2 of [RFC2679].

Accuracy of NetProbe measurements is usually limited by NTP synchronization performance (which is typically taken as ~+/-1ms error or greater), although the installation used in this testing often exhibits errors much less than typical for NTP. The primary stratum 1 NTP server is closely located on a sparsely utilized network management LAN, thus it avoids many concerns raised in Section 10 of[RFC2330] (in fact, smooth adjustment, long-term drift analysis and compensation, and infrequent adjustment all lead to stability during measurement intervals, the main concern).

The resolution of the reported results is 1us (us = microsecond) in the version of NetProbe tested here, which contributes to at least +/-1us error.

NetProbe implements a time-keeping sanity check on sending and receiving time-stamping processes. When the significant process interruption takes place, individual test packets are flagged as possibly containing unusual time errors, and are excluded from the sample used for all "time" metrics.

We performed a NetProbe calibration of the type described in Section 3.7.3 of [RFC2679], using 64 Byte packets over a cross-connect cable. The results estimate systematic and random components of the sum of the Hsource + Hdest errors or uncertainties. In a test with 300 singletons conducted over 30 seconds (periodic sample with 100ms spacing), the median is the systematic error and the remaining variability is the random error. One set of results is tabulated below:

(Results from the "R" software environment for statistical computing and graphics - http://www.r-project.org/)

```
> summary(XD4CAL)
      CAL1            CAL2             CAL3
 Min.   : 89.0   Min.   : 68.00   Min.   : 54.00
 1st Qu.: 99.0   1st Qu.: 77.00   1st Qu.: 63.00
 Median :110.0   Median : 79.00   Median : 65.00
 Mean   :116.8   Mean   : 83.74   Mean   : 69.65
 3rd Qu.:127.0   3rd Qu.: 88.00   3rd Qu.: 74.00
 Max.   :205.0   Max.   :177.00   Max.   :163.00
>
```

NetProbe Calibration with Cross-Connect Cable, one-way delay values in microseconds (us)

The median or systematic error can be as high as 110 us, and the range of the random error is also on the order of 116 us for all streams.

Also, anticipating the Anderson-Darling K-sample (ADK) comparisons to follow, we corrected the CAL2 values for the difference between means between CAL2 and CAL3 (as specified in [I-D.ietf-ippm-metrictest]), and found strong support for the (Null Hypothesis that) the samples are from the same distribution (resolution of 1 us and alpha equal 0.05 and 0.01)

```
> XD4CVCAL2 <- XD4CAL$CAL2 - (mean(XD4CAL$CAL2)-mean(XD4CAL$CAL3))
> boxplot(XD4CVCAL2,XD4CAL$CAL3)
> XD4CV2_ADK <- adk.test(XD4CVCAL2, XD4CAL$CAL3)
> XD4CV2_ADK
Anderson-Darling k-sample test.

Number of samples:  2
Sample sizes: 300 300
Total number of values: 600
Number of unique values: 97

Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.75896

T = (Anderson Darling Criterion - mean)/sigma

Null Hypothesis: All samples come from a common population.

                      t.obs P-value extrapolation
not adj. for ties  0.71734 0.17042             0
adj. for ties     -0.39553 0.44589             1
>
```

4.2. Perfas Error and Type-P

Perfas+ is configured to use GPS synchronisation and uses NTP synchronization as a fall-back or default. GPS synchronisation worked throughout this test with the exception of the calibration stated here (one implementation was NTP synchronised only). The time stamp accuracy typically is 0.1 ms.

The resolution of the results reported by Perfas+ is 1us (us = microsecond) in the version tested here, which contributes to at least +/-1us error.

```
Port    5001 5002 5003
Min.    -227 -226  294
Median  -169 -167  323
Mean    -159 -157  335
Max.       6  -52  376
s        102  102   93
```

Perfas Calibration with Cross-Connect Cable, one-way delay values in microseconds (us)

The median or systematic error can be as high as 323 us, and the range of the random error is also less than 232 us for all streams.

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) comparisons, the required confidence factor for the cross-implementation comparisons SHALL be the smallest of:

- 0.95 confidence factor at 1ms resolution, or
- the smallest confidence factor (in combination with resolution) of the two same-implementation comparisons for the same test conditions.

A constant time accuracy error of as much as +/-0.5ms MAY be removed from one implementation’s distributions (all singletons) before the ADK comparison is conducted.

A constant propagation delay error (due to use of different sub-nets between the switch and measurement devices at each location) of as much as +2ms MAY be removed from one implementation’s distributions (all singletons) before the ADK comparison is conducted.

For comparisons involving the mean of a sample or other central statistics, the limits on both the time accuracy error and the propagation delay error constants given above also apply.

6. Tests to evaluate RFC 2679 Specifications

This section describes some results from real-world (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 slightly modified from the original procedures contained in Appendix A.1 of [I-D.ietf-ippm-metrictest]. The

modifications include the use of the mean statistic for comparisons.

Note that there are only five instances of the requirement term "MUST" in [RFC2679] outside of the boilerplate and [RFC2119] reference.

6.1. One-way Delay, ADK Sample Comparison - Same & Cross Implementation

This test determines if implementations produce results that appear to come from a common delay distribution, as an overall evaluation of Section 4 of [RFC2679], "A Definition for Samples of One-way Delay". 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 3 and 4 of [RFC2679].

By testing the extent to which the distributions of one-way delay singletons from two implementations of [RFC2679] appear to be from the same distribution, we economize on comparisons, because comparing a set of individual summary statistics (as defined in Section 5 of [RFC2679]) would require another set of individual evaluations of equivalence. Instead, we can simply check which statistics were implemented, and report on those facts.

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 delay singletons with 2 or more implementations, using identical options and network emulator settings (if used).

3. Measure a sample of one-way delay singletons with *four* instances of the *same* implementations, using identical options, noting that connectivity differences SHOULD be the same as for the cross implementation testing.

4. Apply the ADK comparison procedures (see Appendix C of [I-D.ietf-ippm-metrictest]) and determine the resolution and confidence factor for distribution equivalence of each same-implementation comparison and each cross-implementation comparison.

5. 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.

6. Apply constant correction factors to all singletons of the sample distributions, as described and limited in Section 5 above.

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:

- IP header + payload = 64 octets
- Periodic sampling at 1 packet per second
- Test duration = 300 seconds (March 29)

The netem emulator was set for 100ms average delay, with uniform delay variation of +/-50ms. 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.

In the result analysis of this section:

- All comparisons used 1 microsecond resolution.
- No Correction Factors were applied.
- The 0.95 confidence factor (1.960 for paired stream comparison) was used.

6.1.1. NetProbe Same-implementation results

A single same-implementation comparison fails the ADK criterion (s1 <-> sB). We note that these streams traversed the test path in opposite directions, making the live network factors a possibility to explain the difference.

All other pair comparisons pass the ADK criterion.

ti.obs (P)	s1	s2	sA
s2	0.25 (0.28)		
sA	0.60 (0.19)	-0.80 (0.57)	
sB	2.64 (0.03)	0.07 (0.31)	-0.52 (0.48)

NetProbe ADK Results for same-implementation

6.1.2. Perfas Same-implementation results

All pair comparisons pass the ADK criterion.

ti.obs (P)	p1	p2	p3
p2	0.06 (0.32)		
p3	1.09 (0.12)	0.37 (0.24)	
p4	-0.81 (0.57)	-0.13 (0.37)	1.36 (0.09)

Perfas ADK Results for same-implementation

6.1.3. One-way Delay, Cross-Implementation ADK Comparison

The cross-implementation results are compared using a combined ADK analysis [ref], where all NetProbe results are compared with all Perfas results after testing that the combined same-implementation results pass the ADK criterion.

When 4 (same) samples are compared, the ADK criterion for 0.95 confidence is 1.915, and when all 8 (cross) samples are compared it is 1.85.

Combination of Anderson-Darling K-Sample Tests.

Sample sizes within each data set:
Data set 1 : 299 297 298 300 (NetProbe)
Data set 2 : 300 300 298 300 (Perfas)
Total sample size per data set: 1194 1198
Number of unique values per data set: 1188 1192
...
Null Hypothesis:
All samples within a data set come from a common distribution.
The common distribution may change between data sets.

NetProbe	ti.obs	P-value	extrapolation
not adj. for ties	0.64999	0.21355	0
adj. for ties	0.64833	0.21392	0
Perfas			
not adj. for ties	0.55968	0.23442	0
adj. for ties	0.55840	0.23473	0

Combined Anderson-Darling Criterion:

	tc.obs	P-value	extrapolation
not adj. for ties	0.85537	0.17967	0
adj. for ties	0.85329	0.18010	0

The combined same-implementation samples and the combined cross-implementation comparison all pass the ADK criteria at P>=0.18 and support the Null Hypothesis (both data sets come from a common distribution).

We also see that the paired ADK comparisons are rather critical. Although the NetProbe s1-sB comparison failed, the combined data set from 4 streams passed the ADK criterion easily.

6.1.4. Conclusions on the ADK Results for One-way Delay

Similar testing was repeated many times in the months of March and April 2011. There were many experiments where a single test stream

from NetProbe or Perfas proved to be different from the others in paired comparisons (even same comparisons). When the out lier stream was removed from the comparison, the remaining streams passed combined ADK criterion. Also, the application of correction factors resulted in higher comparison success.

We conclude that the two implementations are capable of producing equivalent one-way delay distributions based on their interpretation of [RFC2679] .

6.2. One-way Delay, Loss threshold, RFC 2679

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 3.5 of [RFC2679], 3rd bullet point and also Section 3.8.2 of [RFC2679].

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 [RFC2679] . This is a simple way to enforce the constant loss threshold envisioned in [RFC2679] (see specific section references 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 Delay, First-bit to Last bit, RFC 2679

This test determines if implementations register the same relative change in delay from one packet size to another, indicating that the first-to-last time-stamping convention has been followed. This test tends to cancel the sources of error which may be present in an implementation.

See Section 3.7.2 of [RFC2679], and Section 10.2 of [RFC2330].

1. configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs, and ideally including a low-speed link (it was not possible to change the link configuration during testing, so the lowest speed link present was the basis for serialization time comparisons).

2. measure (average) one-way delay with 2 or more implementations, using identical options and equal size small packets (64 octet IP header and payload)

3. maintain the same path with additional emulated 100 ms one-way delay

4. measure (average) one-way delay with 2 or more implementations, using identical options and equal size large packets (500 octet IP header and payload)

5. observe that the increase measured between steps 2 and 4 is equivalent to the increase in ms expected due to the larger serialization time for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

The common parameters used for tests in this section are:

- IP header + payload = 64 octets

- Periodic sampling at l packet per second

- Test duration = 300 seconds total (April 12)

The netem emulator was set to add constant 100ms delay.

6.3.1. NetProbe and Perfas Results for Serialization

When the IP header + payload size was increased from 64 octets to 500 octets, there was a delay increase observed.

Mean Delays in us
NetProbe

Payload	s1	s2	sA	sB
500	190893	191179	190892	190971
64	189642	189785	189747	189467
Diff	1251	1394	1145	1505

Perfas

Payload	p1	p2	p3	p4
500	190908	190911	191126	190709
64	189706	189752	189763	190220
Diff	1202	1159	1363	489

Serialization tests, all values in microseconds

The typical delay increase when the larger packets were used was 1.1 to 1.5 ms (with one outlier). The typical measurements indicate that a link with approximately 3 Mbit/s capacity is present on the path.

Through investigation of the facilities involved, it was determined that the lowest speed link was approximately 45 Mbit/s, and therefore the estimated difference should be about 0.077 ms. The observed differences are much higher.

The unexpected large delay difference was also the outcome when testing serialization times in a lab environment, using the NIST Net Emulator and NetProbe [ref to earlier lab tests].

6.3.2. Conclusions for Serialization

Since it was not possible to confirm the estimated serialization time increases in field tests, we resort to examination of the implementations to determine compliance.

NetProbe performs all time stamping above the IP-layer, accepting that some compromises must be made to achieve extreme portability and measurement scale. Therefore, the first-to-last bit convention is supported because the serialization time is included in the one-way delay measurement, enabling comparison with other implementations.

Perfas >>>>>>>>>>>>>>> TBD

6.4. One-way Delay, Difference Sample Metric (Lab)

This test determines if implementations register the same relative increase in delay from one measurement to another under different delay conditions. This test tends to cancel the sources of error which may be present in an implementation.

This test is intended to evaluate measurements in sections 3 and 4 of [RFC2679].

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 (average) one-way delay with 2 or more implementations, using identical options

3. configure the path with X+Y ms one-way delay

4. repeat measurements

5. observe that the (average) increase measured in steps 2 and 4 is ~Y ms for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

In this test, X=1000ms and Y=1000ms.

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.

6.4.1. NetProbe results for Differential Delay

Average pre-increase delay, microseconds	1089868.0
Average post 1s additional, microseconds	2089686.0
Difference (should be ~= Y = 1s)	999818.0

Average delays before/after 1 second increase

The NetProbe implementation observed a 1 second increase with a 182 microsecond error (assuming that the netem emulated delay difference is exact).

We note that this differential delay test has been run under lab conditions and published in prior work [ref to "advance metrics" draft]. The error was 6 microseconds.

6.4.2. Perfas results for Differential Delay

Average pre-increase delay, microseconds	1089794.0
Average post 1s additional, microseconds	2089801.0
Difference (should be ~= Y = 1s)	1000007.0

Average delays before/after 1 second increase

The Perfas implementation observed a 1 second increase with a 7 microsecond error.

6.4.3. Conclusions for Differential Delay

Again, the live network conditions appear to have influenced the results, but both implementations measured the same delay increase within their calibration accuracy.

6.5. Implementation of Statistics for One-way Delay

The ADK tests the extent to which the sample distributions of one-way delay singletons from two implementations of [RFC2679] appear to be from the same overall distribution. By testing this way, we economize on the number of comparisons, because comparing a set of individual summary statistics (as defined in Section 5 of [RFC2679]) would require another set of individual evaluations of equivalence. Instead, we can simply check which statistics were implemented, and report on those facts, noting that Section 5 of [RFC2679] does not specify the calculations exactly, and gives only some illustrative examples.

	NetProbe	Perfas
5.1. Type-P-One-way-Delay-Percentile	yes	no
5.2. Type-P-One-way-Delay-Median	yes	no
5.3. Type-P-One-way-Delay-Minimum	yes	yes
5.4. Type-P-One-way-Delay-Inverse-Percentile	no	no

Implementation of Section 5 Statistics

5.1. Type-P-One-way-Delay-Percentile 5.2. Type-P-One-way-Delay-Median 5.3. Type-P-One-way-Delay-Minimum 5.4. Type-P-One-way-Delay-Inverse-Percentile

7. Security Considerations

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

8. IANA Considerations

This memo makes no requests of IANA, and hopes that IANA will be as accepting of our new computer overlords as the authors intend to be.

9. 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 with Ganga Maguluri.

10. References

10.1. Normative References

[I-D.ietf-ippm-metrictest] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IPPM standard advancement testing", draft-ietf-ippm-metrictest-02 (work in progress), March 2011.

[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.

[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.

10.2. Informative References

[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.

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

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
University of Applied Sciences Darmstadt
Birkenweg 8 Department EIT
Darmstadt, 64295
Germany

Phone:
Email: matthias.wieser@stud.h-da.de

Network Working Group
Internet-Draft
Updates: 5357 (if approved)
Intended status: Standards Track
Expires: January 1, 2012

A. Morton
L. Ciavattone
AT&T Labs

June 30, 2011

TWAMP Burst Rate Measurement Features
draft-morton-ippm-twamp-rate-00

Abstract

This memo describes two rate-measurement features for the core specification of TWAMP - the Two-Way Active Measurement Protocol: an optional capability where the reflector host responds with a controlled burst of test-session packets (instead of a single packet), and an optional test mode that requires the responder to measure a burst of test packets and communicate the results in truncated packet(s). Both features add the ability to control packet size in the tested direction, enabling asymmetrical packet size testing.

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 1, 2012.

Copyright Notice

Table of Contents

1. Introduction

TWAMP - the Two-Way Active Measurement Protocol [RFC5357] is an extension of the One-way Active Measurement Protocol, OWAMP [RFC4656]. The TWAMP specification gathered wide review as it was deployed, resulting in recommendations for new features.

This memo describes two closely-related features for TWAMP. When measuring packet delivery rate to end-systems, unique control and measurement capabilities become useful, especially when the path tested includes asymmetrical link speeds (as are often deployed in consumer Internet access services).

One feature is the OPTIONAL capability for the responder host to return a controlled burst of test-session packets (instead of a single packet).

Another is an optional sender packet format that requires the responder to measure a burst of test packets and communicate the results in a single packet.

Both features add the ability to control packet size in each direction, enabling asymmetrical packet size testing. Although TWAMP [RFC5357] recommends padding truncation to achieve symmetrical sizes (to compensate for the Session-Reflector's larger test packet header), these features configure test packet sizes when the test session is requested using the TWAMP-Control protocol.

We note that [draft-baillargeon-ippm-twamp-value-added-octets-01.txt] addresses a similar measurement problem, but places different requirements on the reflector host and does not include the asymmetrical size aspect.

This memo is an update to the TWAMP core protocol specified in [RFC5357]. Measurement systems are not required to implement the features described in this memo to claim compliance with [RFC5357].

Throughout this memo, the bits marked MBZ (Must Be Zero) MUST be set to zero by senders and MUST be ignored by receivers. Also, the HMAC (Hashed Message Authentication Code) MUST be calculated as defined in Section 3.2 of [RFC4656].

2. Purpose and Scope

The purpose of this memo is to define two OPTIONAL closely-related features for TWAMP [RFC5357]. The features enhance the TWAMP responder's capabilities to perform a simple operations on test

packets, and the capability to demand asymmetrical size TWAMP-Test packets.

The scope of the memo is limited to specifications of the following features:

- Burst Generation: the capability of the Session-Reflector to generate a burst of packets for return to the Session-Sender, and the corresponding TWAMP-Control messages to activate the capability between compliant hosts.

- Burst Measurement: the capability of the Session-Reflector to measure a burst of packets from the Session-Sender, report the key information (receive timestamps) in the response packet(s), and the corresponding TWAMP-Control messages to activate the capability between compliant hosts.

- Asymmetrical Size: the capability to ensure that TWAMP-Test protocol uses a specific packet size in each direction. This feature is combined with the Burst features, and essentially adds a third simple capability when the Burst size = 1.

This memo extends the modes of operation through assignment of two new values in the Modes Field (see section 3.1 of[RFC4656] for the format of the Server Greeting message), while retaining backward compatibility with the core TWAMP [RFC5357] implementations. The two new values correspond to the two features defined in this memo.

When the Server and Control-Client have agreed to use the Burst Generation mode during control connection setup, then the Control-Client, the Server, the Session-Sender, and the Session-Reflector MUST all conform to the requirements of that mode, as identified below.

When the Server and Control-Client have agreed to use the Burst Measurement mode during control connection setup, then the Control-Client, the Server, the Session-Sender, and the Session-Reflector MUST all conform to the requirements of that mode, as identified below.

3. TWAMP Control Extensions

TWAMP-Control protocol [RFC5357] uses the Modes Field to identify and select specific communication capabilities, and this field is a recognized extension mechanism. The following sections describe two such extensions.

3.1. Connection Setup with New Features

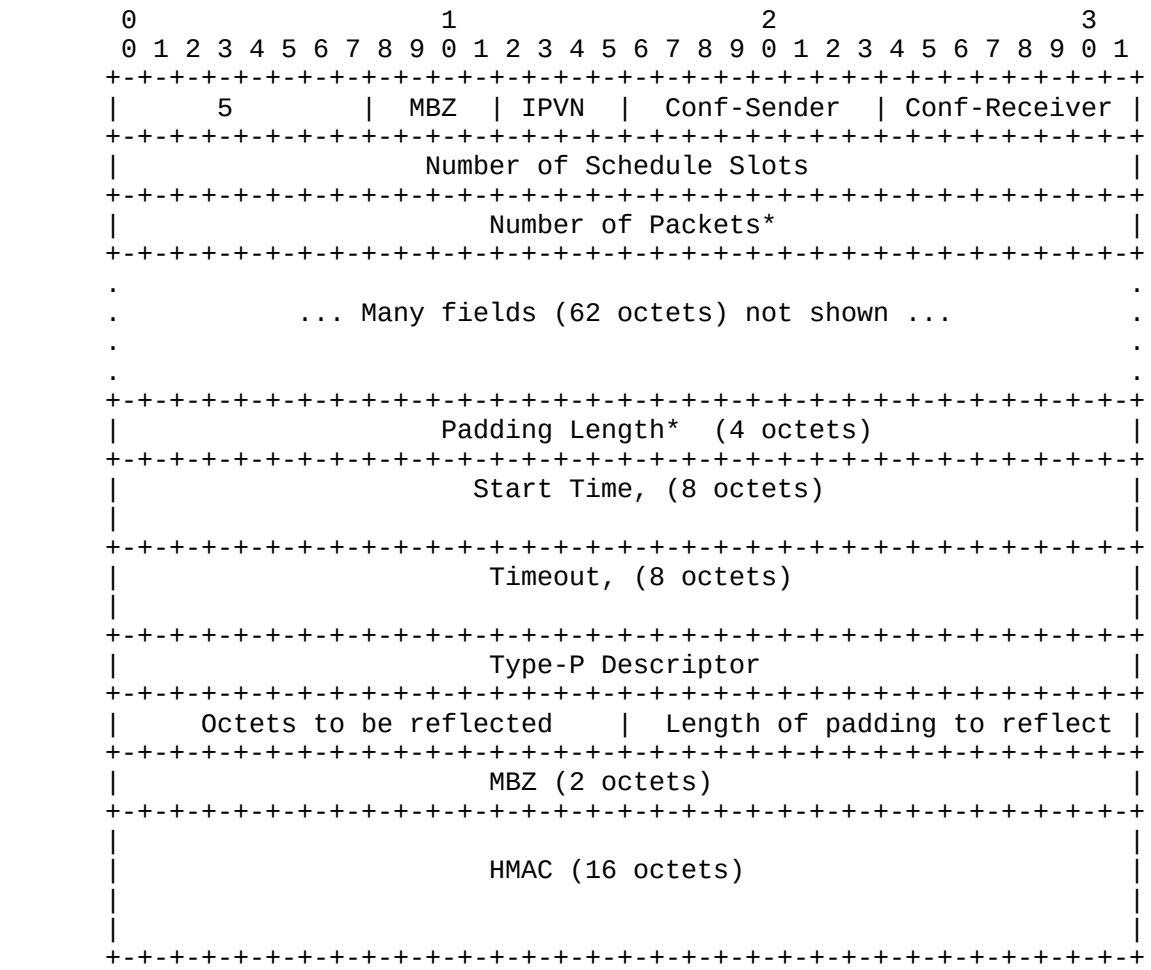
TWAMP connection establishment follows the procedure defined in section 3.1 of [RFC4656] and section 3.1 of [RFC5357]. The new features require two new bit positions (and values). See the IANA section for details on the assigned values and bit positions.

The Server sets one or both of the new bit positions in the Modes Field of the Server Greeting message to indicate its capabilities and willingness to operate in either of these modes if desired.

If the Control-Client intends to operate all test sessions invoked with this control connection using one of the new modes, it MUST set the Mode Field bit corresponding to each function in the Setup Response message. With this and other extensions, the Control-Client MAY set multiple Mode Field bits in the Setup Response message, but these new features are mutually exclusive, and MUST NOT be used together.

3.2. Burst Generation: Request-TW-Session Packet Format

The bits designated for the Burst Generation feature in the Request-TW-Session command are as shown in the packet format below.



* = re-interpreted field

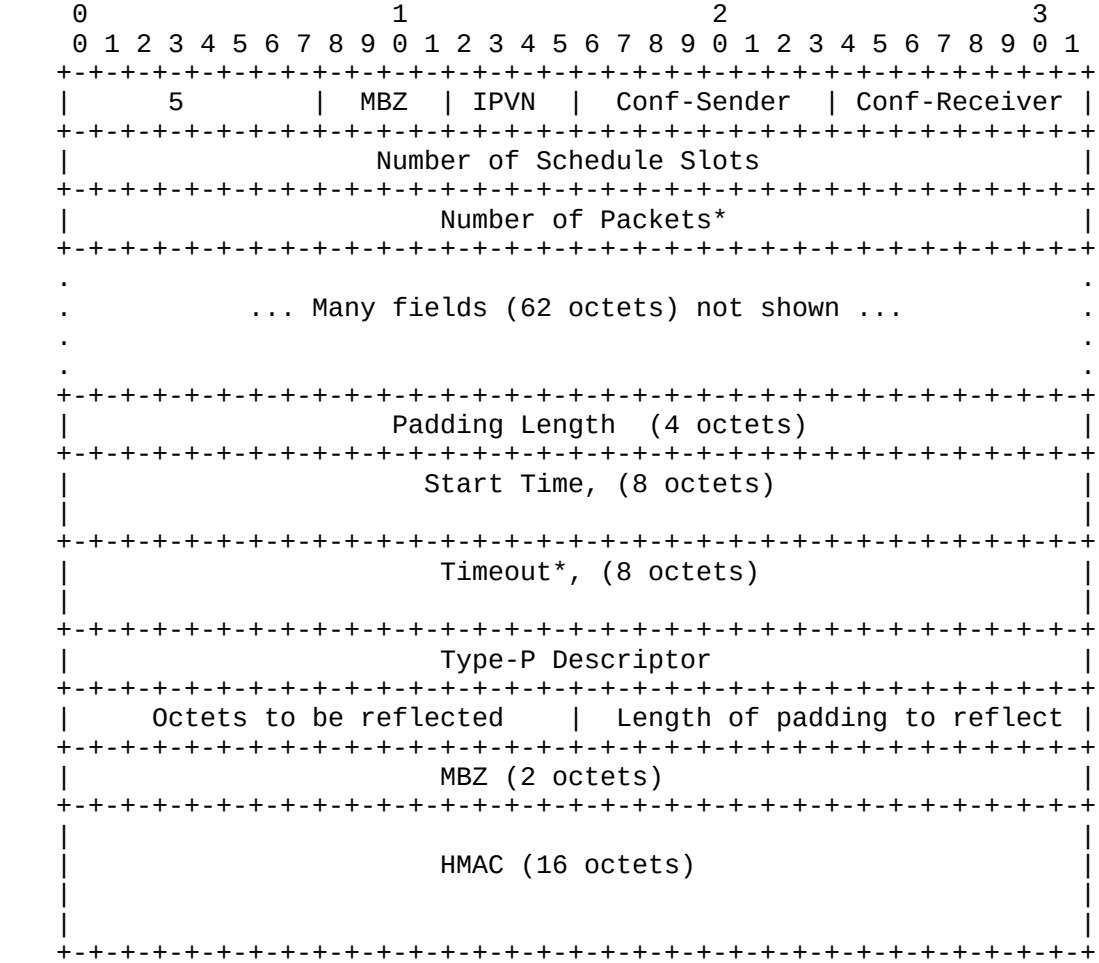
Two re-interpreted fields appear in the Request-TW-Session command when using Burst Generation mode:

1. Number of Packets: In this mode, re-interpreted as the number of packets that the Session-Reflector MUST generate in each Burst.

2. Packet Padding Length: In the mode, re-interpreted as the number of octets the Session-Reflector MUST append to the Test packet header of each packet it generates as part of the burst. The Session-Reflector MUST NOT assume that the Session-Sender will use any packet padding, and MUST be prepared to generate the padding itself.

3.3. Burst Measurement: Request-TW-Session Packet Format

The bits designated for the Burst Generation feature in the Request-TW-Session command are as shown in the packet format below.



* = re-interpreted field

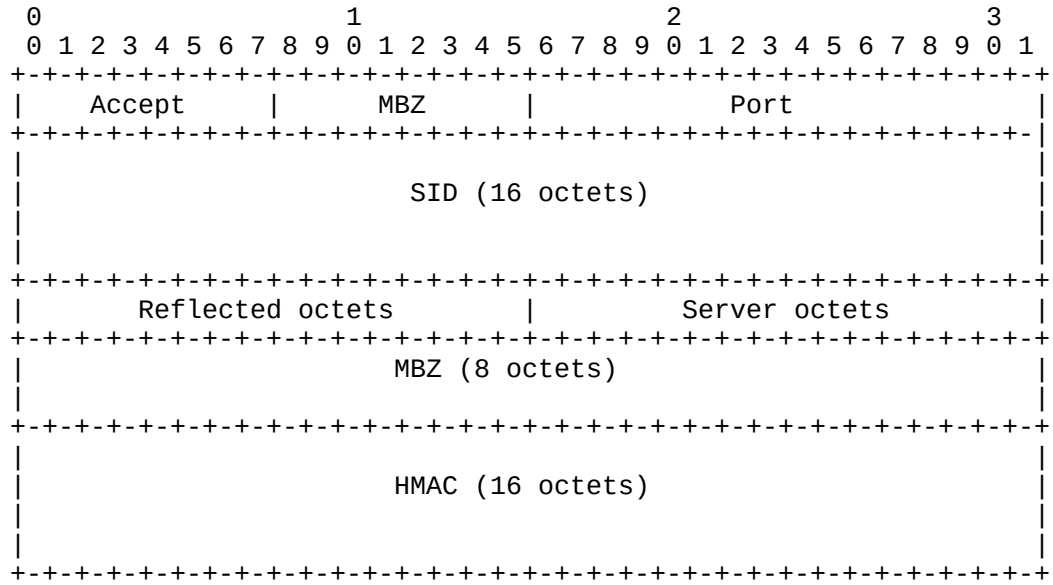
Two re-interpreted fields appear in the Request-TW-Session command when using Burst Measurement mode:

1. Number of Packets: In this mode, re-interpreted as the number of packets that the Session-Reflector MUST expect to measure as part of each Burst.

2. Timeout: In this mode, re-interpreted as the time to wait for all packets in a burst to arrive, expressed in the existing timestamp format used in TWAMP and OWAMP. In the case of lost packets, the Session-Reflector is commanded to wait through this time-out for packets in a burst to arrive.

3.4. Burst Gen and Meas: Accept Session Packet Format

The Accept Session command for the Burst feature is as shown in the packet format below (assuming the Reflect Octets feature is also in use).



3.5. Burst Gen and Meas: Stopping Test Sessions

The Control-Client SHALL stop in-progress test sessions using any standardized methods, including section 3.8 of [RFC5357] or the optional capability of [RFC5938].

3.6. Additional considerations

The value of the Modes Field sent by the Server in the Server Greeting message is the bit-wise OR of the mode values that it is willing to support during this session.

We note that Burst Generation and Measurement features are incompatible with each other, and with the Symmetrical Size feature

described in [RFC6038], and MUST NOT be used in combination with those features.

With the publication of this memo as an RFC, the last 9 bit positions of the Modes 32-bit Field are used. A Control-Client conforming to this extension of [RFC5357] MAY ignore the values in the higher bits of the Modes Field, or it MAY support other features that are communicated in those bit positions. The other bits are available for future protocol extensions.

4. Burst Generation in TWAMP Test

The TWAMP test protocol is similar to the OWAMP [RFC4656] test protocol with the exception that the Session-Reflector transmits test packets to the Session-Sender in response to each test packet it receives. The Burst Generation feature modifies the behavior of TWAMP section 4[RFC5357]. This mode requires the Session-Sender to send a Burst-Initiation packet, and the Session-Reflector generates test session packets according to the configuration agreed using the TWAMP-Control protocol.

4.1. Sender Behavior

This section describes extensions to the behavior of the TWAMP Session-Sender.

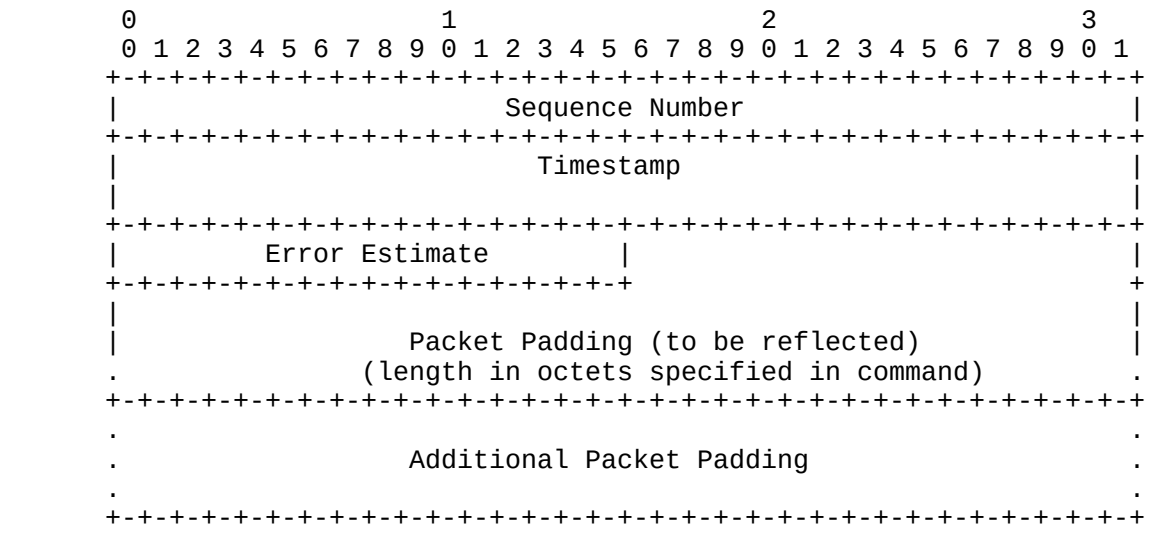
4.1.1. Packet Timings

The Send Schedule is not utilized in TWAMP, and this is unchanged in this memo.

4.1.2. Packet Formats and Contents

The Session-Sender packet format and content follow the same procedure and guidelines as defined in section 4.1.2 of [RFC4656] (as indicated in section 4.1.2 of TWAMP [RFC5357]).

This mode uses the original TWAMP-Test Packet Padding Field (see section 4.1.2 of [RFC4656]), or can be used with Reflect Octets feature as shown below for unauthenticated mode:



Burst Initiation Packet Format

The Sequence Number, Timestamp, and Error Estimate fields are the same as specified in section 4.1.2 of [RFC4656] in OWAMP.

We note that the format of the Burst Initiation packet has not been changed from the usual Session-Sender test packet format, to simplify adoption.

4.2. Reflector Behavior

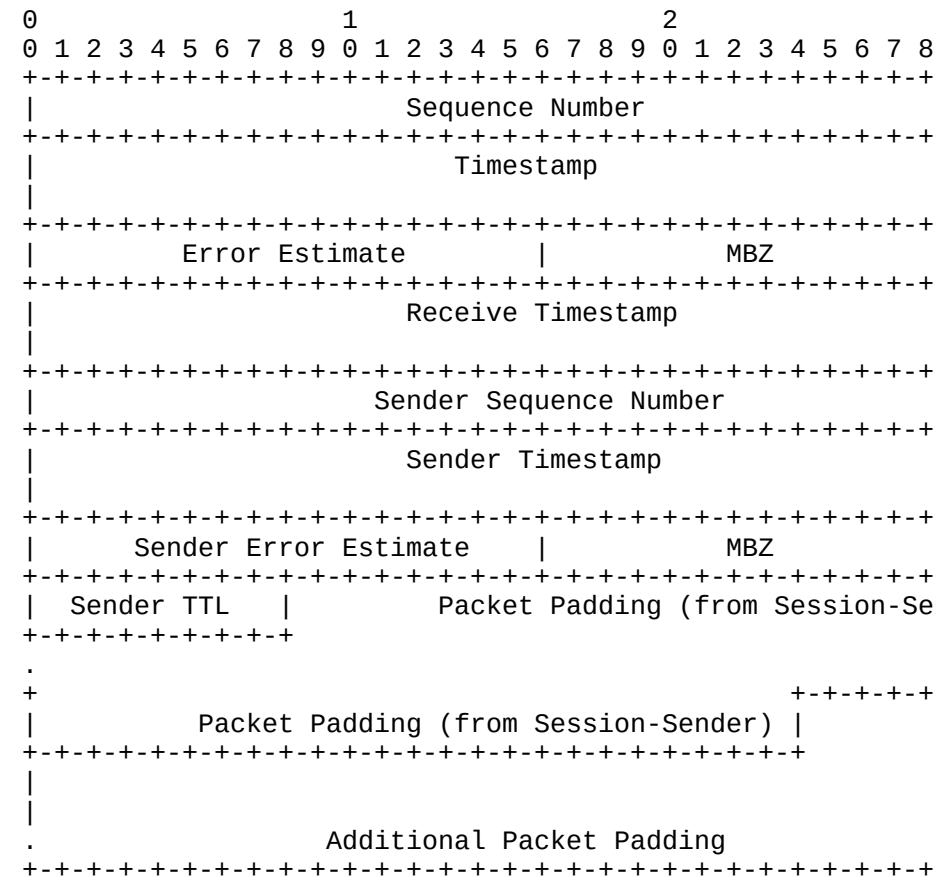
The TWAMP Reflector differs significantly from the procedures and guidelines in section 4.2 of [RFC5357]. The following new functions MUST be performed:

- o Recognition of the function of the Burst Initiation Packet used in this mode.

- o Generation of the required burst of test session packets, according to the configuration agreed in Request-TW-Session command, with the agreed number of packets in each burst and size of each packet in the burst.

4.2.1. Session-Reflector Burst Packet Format and Contents

The Burst Generation feature retains the usual Reflector packet fields, as shown below. When the Burst Generation mode is selected, the Session-Reflector SHALL use the following TWAMP-Test Packet Format in Unauthenticated mode (shown with Reflect Octets feature activated):



Section 4.2.1 of [RFC5357] describes the above fields as used in TWAMP, with one exception.

The Sequence Number field SHALL indicate the sequence number of each packet sent throughout the test session. The Sequence Number SHALL be increased by 1 for each packet. The initial Sequence Number SHALL be 0.

When one burst is complete, the Sequence Numbers SHALL continue to increment by 1 in the packets generated in response to the next burst.

The total Packet Padding octets SHALL have the length specified in the TWAMP-Control request for the appropriate test session. The Session-Reflector MAY need to generate its own packet padding, if the Burst Request packet does not include this field (or contains insufficient padding).

In any case, the Session-Reflector MAY re-use the Sender's Packet Padding (since the requirements for padding generation are the same for each) when possible.

The Session-Reflector SHALL send a series of TWAMP-Test Packets in response to reception of the Burst Initiation Packet, according to the configuration agreed in the Request-TW-Session command (number of packets and padding), and as immediately as possible. The Session-Reflector SHALL send all packets in a burst as close to back-to-back as possible (recognizing that lower layers may have spacing requirements that take precedence).

5. Burst Measurement in TWAMP Test

The Burst Measurement feature modifies the behavior of TWAMP section 4[RFC5357]. This mode requires the Session-Sender to send a Burst of test packets, and the Session-Reflector measures the burst of packets and reports the results in the Burst Response packet format(s), as described below.

5.1. Sender Behavior

This section describes extensions to the behavior of the TWAMP Session-Sender.

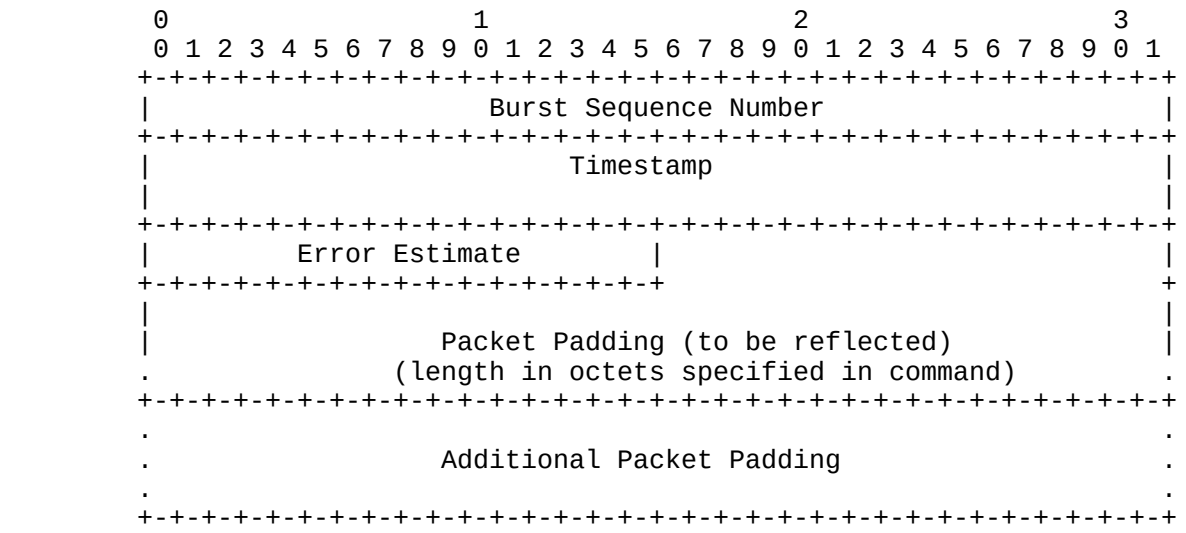
5.1.1. Packet Timings

The Session-Sender SHALL send all packets in a burst as close to back-to-back as possible (recognizing that lower layers may have spacing requirements that take precedence).

5.1.2. Packet Formats and Contents

The Session-Sender packet format and content SHALL comply with that defined in section 4.1.2 of [RFC4656] (as indicated in section 4.1.2 of TWAMP [RFC5357]).

This mode uses the original TWAMP-Test Packet Padding Field (see section 4.1.2 of [RFC4656]), or can be used with Reflect Octets feature as shown below for unauthenticated mode:



Session-Sender Burst Test Packet Format

The Burst Sequence Number field SHALL indicate the number of each burst. The Burst Sequence Number SHALL be increased by 1 for each burst, and remain the same for each packet in a burst. The initial number SHALL be 0.

When one burst is complete, the Burst Sequence Number used in the all packets of the next burst SHALL be increased by 1.

5.2. Reflector Behavior

The TWAMP Reflector differs slightly from the procedures and guidelines in section 4.2 of [RFC5357]. The following new functions MUST be performed:

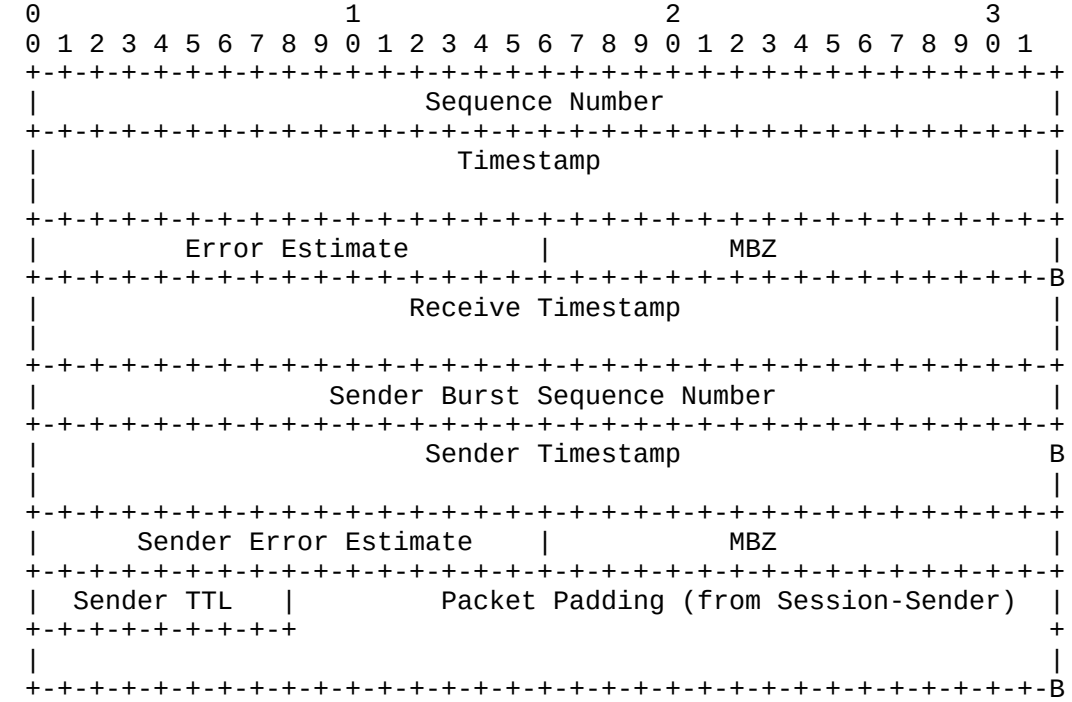
- o Recognition of the function of the Session-Sender Burst Test Packet Format used in this mode.
- o Processing the required bursts of test session packets, according to the configuration agreed in Request-TW-Session command, with the agreed length of the burst in packets and size of each packet in the burst, and the agreed Burst Time-out.
- o Response with an abbreviated Session-Reflector test packet as described below. For discussion, we will call this the 1-to-1 response.
- o OR - Response with the new Burst Measurement Response packet described below. For discussion, we will call this the

accumulated response.

We seek feedback from the IPPM working group on which of these two alternatives is preferable.

5.2.1. Session-Reflector Burst Measurement Response Packet Format and Contents

The Burst Measurement feature specifies a standard Session-Reflector packet to communicate the results, as shown below. When the Burst measurement mode is selected, the Session-Sender SHALL use the following Burst Measurement Response packet Format in Unauthenticated mode (shown with Reflect Octets feature also in use):



Session-Reflector Measurement Packet (1-to-1 response)

Section 4.2.1 of [RFC5357] describes the fields in the 1-to-1 response packet above; they are the same as used in TWAMP. The main difference is that Packet Padding SHALL be truncated on a 16 octet-word boundary, returning the minimum information to the Session-Sender.

All Timestamps SHALL be formatted according to the precedent set in section 4.1.2 of [RFC4656], which is to use [RFC1305] (and updated

version), as follows:

"The first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the next 32 bits represent the fractional part of a second that has elapsed since then."

The Session-Reflector MUST truncate the Sender's Packet Padding, unless the Reflect Octets feature is also active in which case the Session_Reflector MAY re-use the Sender's Packet Padding (since the requirements for padding generation are the same for each) to reach a word boundary.

The Sender Timestamp field SHALL have the sender's timestamp from each packet received in the burst.

In 1-to-1 response mode, the Session-Reflector SHALL send a Session-Reflector Measurement Packet in response to every Session-Sender packet received, and as immediately as possible.

==========

In the accumulated response alternative, the Session-Reflector creates and holds all packet headers described above in a buffer, and sends them all at once in a single Session-Reflector test packet. The length of the burst and the path MTU MUST be coordinated to avoid fragmentation.

The first Session-Sender packet to arrive with a previously unseen Burst Sequence Number SHALL be designated as the "First" packet in that burst, and its timestamp is used in processing below.

As subsequent packets arrive, Session-Reflector SHALL:

- Maintain a count of packets with the same Burst Sequence Number (one burst).
- Time stamp each packet as it arrives and store the time stamp in a response packet structure with all fields complete, as in the 1-to-1 alternative.

When

- The count of packets with the same Burst Sequence Number equals the agreed Burst Length, OR
- The agreed Timeout expires (computed by a the time to the "First" Packet Timestamp), OR

- o The Burst Sequence Number increases from previous packets (indicating a new Burst is in progress),

then the current burst is determined to be complete.

When the Burst is complete, the Session-Reflector SHALL terminate the current burst processing as described above and send the Burst Measurement Response Packet to the Session-Sender as immediately as possible.

In Accumulated Response, the Burst Measurement Response Packet is a single packet with the concatenation of all previously-generated response packet formats in the information field.

6. Special Case of One-packet Bursts

When the Number of Packets field in the Request-TW-Session command equals 1, then the Burst Generation and Measurement modes are reduced to test sessions with controlled, asymmetrical packet sizes. A minimal size packet travels in one direction, and the measured direction uses a packet with all Packet Padding specified in the Request-TW-Session command.

7. Security Considerations

These extended modes of operation do not appear to permit any new attacks on hosts communicating with core TWAMP [RFC5357].

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

8. IANA Considerations

This memo adds two modes to the IANA registry for the TWAMP Modes Field, and describes behavior when the new modes are used. This field is a recognized extension mechanism for TWAMP.

8.1. Registry Specification

IANA has created a TWAMP-Modes registry (as requested in [RFC5618]). TWAMP-Modes are specified in TWAMP Server Greeting messages and Set-up Response messages, as described in section 3.1 of [RFC5357], consistent with section 3.1 of [RFC4656], and extended by this memo. Modes are indicated by setting bits in the 32-bit Modes field that

correspond to values in the Modes registry. For the TWAMP-Modes registry, we expect that new features will be assigned increasing registry values that correspond to single bit positions, unless there is a good reason to do otherwise (more complex encoding than single bit positions may be used in the future, to access the 2^32 value space).

8.2. Registry Contents

TWAMP Modes Registry is recommended to be augmented as follows:

Value	Description	Semantics Definition
xxx	Burst Generation Capability	this memo, section 3.1 new bit position (X)
yyy	Burst Measurement	this memo, section 3.1 new bit position (Y)

>>>IANA: change xxx, yyy, X, Y, and RFC???? to the assigned values

The suggested values are

X=7, xxx=128

Y=8, yyy=256 <<<<

9. Acknowledgements

The authors thank folks for review and comment.

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.

[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.

10.2. Informative References

Authors' Addresses

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

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

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

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