Technical Report

TR-452.2
Quality Attenuation Measurements using Active Test Protocols

Issue 1
Issue Date: November 2022

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report has been approved by members of the Forum. This Technical Report is subject to change. This Technical Report is owned and copyrighted by the Broadband Forum, and all rights are reserved. Portions of this Technical Report may be owned and/or copyrighted by Broadband Forum members.

Intellectual Property

Recipients of this Technical Report are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of this Technical Report, or use of any software code normatively referenced in this Technical Report, and to provide supporting documentation.

Terms of Use

1. License

Broadband Forum hereby grants you the right, without charge, on a perpetual, non-exclusive and worldwide basis, to utilize the Technical Report for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, products complying with the Technical Report, in all cases subject to the conditions set forth in this notice and any relevant patent and other intellectual property rights of third parties (which may include members of Broadband Forum). This license grant does not include the right to sublicense, modify or create derivative works based upon the Technical Report except to the extent this Technical Report includes text implementable in computer code, in which case your right under this License to create and modify derivative works is limited to modifying and creating derivative works of such code. For the avoidance of doubt, except as qualified by the preceding sentence, products implementing this Technical Report are not deemed to be derivative works of the Technical Report.

2. NO WARRANTIES

THIS TECHNICAL REPORT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT AND ANY IMPLIED WARRANTIES ARE EXPRESSLY DISCLAIMED. ANY USE OF THIS TECHNICAL REPORT SHALL BE MADE ENTIRELY AT THE USER'S OR IMPLEMENTER'S OWN RISK, AND NEITHER THE BROADBAND FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY USER, IMPLEMENTER, OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS TECHNICAL REPORT, INCLUDING BUT NOT LIMITED TO, ANY CONSEQUENTIAL, SPECIAL, PUNITIVE, INCIDENTAL, AND INDIRECT DAMAGES.

3. THIRD PARTY RIGHTS

Without limiting the generality of Section 2 above, BROADBAND FORUM ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE TECHNICAL REPORT IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE TECHNICAL REPORT, BROADBAND FORUM TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

All copies of this Technical Report (or any portion hereof) must include the notices, legends, and other provisions set forth on this page.

Issue History

Issue Number	Approval Date	Publication Date	Issue Editor	Changes
1	1 November 2022	1 November 2022	Peter Thompson, Predictable Network Solutions Ltd.	Original

Comments or questions about this Broadband Forum Technical Report should be directed to info@broadband-forum.org.

Editor: Peter Thompson, Predictable Network Solutions Ltd.

Work Area Director(s): David Sinicrope, Ericsson

Project Stream Leader(s): Gregory Mirsky, Ericsson

Table of Contents

Table of Figures

Table of Tables

Executive Summary

While a general approach to measuring Quality Attenuation (∆Q) has been described in TR-452.1 [3], requiring new capabilities not already present in network nodes constrains the capacity to deploy the technique. Making deployment 'frictionless' means enabling simple and ubiquitous abilities to deploy and run measurements. The aim is to support measurements to and from as many locations as possible across existing and new networks with the minimum change to existing software and/or hardware.

This document specifies how to perform ∆Q measurements using active single-sided two-way measurement protocols that are already in use. It specifies:

- In what context and circumstances active test protocols can be used to measure ∆Q;
- What specific or optional features of particular active test protocols are required to measure ∆Q;
- How to configure active test protocols components in order to measure ∆Q.
    - Specifically, TR-390 and TR-390 issue 2 (TWAMP Light and STAMP)

Informative Appendices discuss issues of timing accuracy, the relationship of this measurement framework to the terminology of TR-304 [6] and an approach to calculating how many test packets are required.

1 Purpose and Scope

1.1 Purpose

Quality Attenuation (ΔQ) measurements have already been made in networks using a variety of platforms and technologies ranging from public cloud compute platforms, PCs/laptops, small micro-computer boxes plugged into end-user broadband routers to rack-mounted servers installed in the network. These approaches have worked well for specific technology investigations and network health checks. However, to reach its full potential it is desirable to be able to cost-effectively make ΔQ observations at multiple nodes/links in the end-to-end broadband connection and on a wider scale.

A general approach to measuring ΔQ has been described in TR-452.1 [3], but where this requires new capabilities not already present in network nodes there is a corresponding constraint in the capacity to deploy the technique. Making deployment 'frictionless' means enabling simple and ubiquitous abilities to deploy and run Quality Attenuation measurements. The aim is to support measurements to and from as many locations as possible across existing and new networks with the minimum change to existing software and/or hardware.

The concept of frictionless deployment is to leverage protocols, technologies and measurement probe capabilities that are already deployed in broadband networks to facilitate ΔQ measurements. This may necessitate some modification or enhancement to these existing capabilities but that is potentially less onerous than deployment of new probes. The objective of this document is to specify how to perform ΔQ measurements using the various active single-sided two-way measurement protocols, including the network and/or test equipment requirements necessary to enable measurement of ΔQ according to TR-452.1 [3] for each member of this class. It specifies how to configure and use each exemplar to perform measurements of ΔQ, including how to deal with the loss of information associated with measurement packet loss.

Single-sided two-way active measurement protocols ('active test protocols' for short) produce 'active metrics' in the terminology of RFC-7799 [11]. They are designed for collection of one-way packet loss and delay metrics as defined by RFC-7679 [15] and RFC-7680 [16] but with the results for both directions (two-way) available at one end of the test (single sided), which is generally the initiator of the test or session sender. This is well aligned with the process of measuring ΔQ, which extends the earlier metrics by:

1. Reporting distributions rather than any averaged values;
2. Varying test packet sizes in order to separate out different components of delay/loss.

1.2 Scope

This Technical Report defines how the active single-sided two-way measurement protocols may be used to perform measurements of ΔQ in accordance with the general framework set out in [3].

This Technical Report specifies:

- In what context and circumstances active test protocols can be used to measure ΔQ;
- What specific or optional features of particular active test protocols are required to measure ΔQ;
- How to configure active test protocols components in order to measure ΔQ.
    - Specifically, TR-390 and TR-390 issue 2 (TWAMP Light and STAMP respectively)

Note that the main body of the text deals with the whole family, while normative annexes deal with specific differences.

2 References and Terminology

2.1 Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [9].

MUST	This word, or the term “REQUIRED”, means that the definition is an absolute requirement of the specification.
MUST NOT	This phrase means that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the term “RECOMMENDED”, means that there could exist valid reasons in particular circumstances to ignore this item, but the full implications need to be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" means that there could exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	This word, or the term “OPTIONAL”, means that this item is one of an allowed set of alternatives. An implementation that does not include this option MUST be prepared to inter-operate with another implementation that does include the option.

2.2 References

The following references are of relevance to this Technical Report. At the time of publication, the editions indicated were valid. All references are subject to revision; users of this Technical Report are therefore encouraged to investigate the possibility of applying the most recent edition of the references listed below.

A list of currently valid Broadband Forum Technical Reports is published at www.broadband-forum.org.

Document	Title	Source	Year
[1] TR-390	Performance Measurement from IP Edge to Customer Equipment using TWAMP Light	BBF	2017
[2] TR-390 issue 2	Performance Measurement from IP Edge to Customer Equipment using TWAMP Light	BBF	2020
[3] TR-452.1	Quality Attenuation Measurement Architecture and Requirements	BBF	2020
[4] TR-143	Enabling Network Throughput Performance Tests and Statistical Monitoring	BBF	2008
[5] TR-145	Multi-service Broadband Network Functional Modules and Architecture	BBF	2012
[6] TR-304	Broadband Access Service Attributes and Performance Metrics	BBF	2015
[7] TR-178 issue 2	Multi-service Broadband Network Architecture and Nodal Requirements	BBF	2017
[8] RFC 5357	Two-Way Active Measurement Protocol (TWAMP)	IEEE	2005

[9]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	IETF	1997
[10]	RFC-8174	Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words	IETF	2017
[11]	RFC-7799	Active and Passive Metrics and Methods (with Hybrid Types In-Between)	IETF	2016
[12]	RFC-4656	A One-way Active Measurement Protocol (OWAMP)	IETF	2006
[13]	RFC-6038	Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features	IETF	2010
[14]	RFC-7594	A Framework for Large-Scale Measurement of Broadband Performance (LMAP)	IETF	2015
[15]	RFC-7679	A One-Way Delay Metric for IP Performance Metrics (IPPM)	IETF	2016
[16]	RFC-7680	A One-Way Loss Metric for IP Performance Metrics (IPPM)	IETF	2016
[17]	RFC 8762	Simple Two-Way Active Measurement Protocol (STAMP)	IETF	2020
[18]	RFC 8972	Simple Two-Way Active Measurement Protocol Optional Extensions	IETF	2021
[19]	STAMP TLV Types registry	STAMP Timestamping Methods sub-registry	IANA	2021

2.3 Definitions

The following terminology is used throughout this Technical Report.

OP	Observation Point
TS	Test Stream
CPE	Customer Premises Equipment.
Stationarity	Stationarity means that the statistical properties of a process generating a time series do not change over time.

2.4 Abbreviations

This Technical Report uses the following abbreviations:

ΔQ	Quality Attenuation.
BNG	Broadband Network Gateway
CDN	Content Delivery Network
CPE	Customer Premises Equipment.
DLM	Dynamic Line Management
GPS	Global Positioning System
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPG	Inter Packet Gap

IRV	Improper Random Variables
ITU-T	International Telecommunications Union – Telecommunication Standardization Sector
MEC	Mobile Edge Computing
NAT	Network Address Translation
NOC	Network Operations Centers
NTP	Network Time Protocol
OP	Observation Point
PDU	Protocol Data Unit
PG	Packet Generator
PR	Packet Reflector
PRO	Predictable Region of Operation
PTP	Precision Time Protocol
RCA	Root Cause Analysis
SDU	Service Data Unit
SRA	Seamless Rate Adaptation
TCP	Transport Control Protocol
TDM	Time-Division Multiplexing
TWAMP	Two-Way Active Measurement Protocol
TR	Technical Report
ULL	Ultra-Low Latency
UX	User eXperience
VoIP	Voice over IP
VNF	Virtual Network Function
vBNG	Virtual Broadband Network Gateway
WA	Work Area
WT	Working Text

3 Technical Report Impact

3.1 Energy Efficiency

TR-452.2 has minimal impact on energy efficiency, although there will be an energy cost associated with performing measurements. It is recommended that the frequency of test packets should be kept low, so that the incremental cost is small.

3.2 Security

TR-452.2 has no impact on security, insofar as it exploits protocols and network node capabilities that are already defined elsewhere. Appropriate best practices should be followed as recommended in the corresponding standards, for example, as specified in TR-390 [1].

Where an additional Collator component is used (as discussed in the section on Observation collation) this needs to be appropriately managed so as to avoid introducing a security risk. Issues relating to the provenance of the data need to be appropriately managed.

3.3 Privacy

TR-452.2 has no impact on privacy.

4 Considerations for test stream generation and observation

TR-452.1 [3] gives a detailed description of the actual measurement requirements (and calculation) for Quality Attenuation and examples of tools that can be used. In summary, we need to be able to transmit streams of test packets that have varying packet sizes and varying departure times (i.e., we can use a "schedule" for them).

Typically, the data rate of the test stream is approximately 30kbit/s and the packet rate can be adjusted within limits defined by the required statistical accuracy. Note that there is a relationship between the packet rate, accuracy of ΔQ measurement and sensitivity to stationarity: at a given packet rate, ΔQ accuracy is improved by aggregating more samples (since it is a statistical measure); on the other hand, the longer we wait before completing a measurement, the less able we are to see whether ΔQ is changing over time. Running multiple simultaneous tests can provide additional information (by measuring multiple network paths concurrently).

The topology of the basic capabilities we would like to construct is illustrated in Figure 1 below:

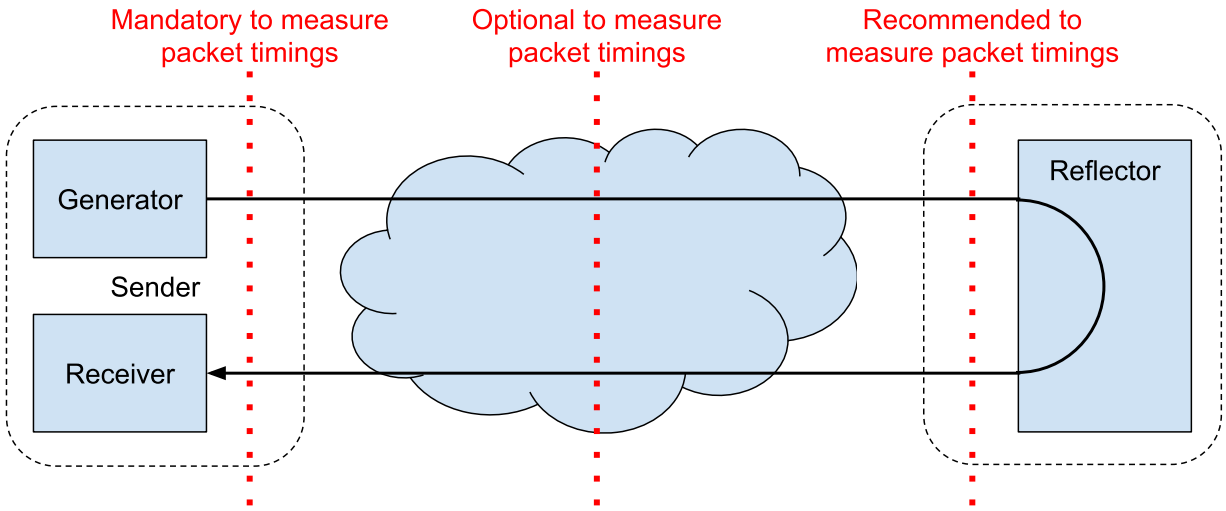

Figure 1 Topology to Inject, Reflect & Measure Low-Rate Test Packets

To get useful results, we need to measure the arrival time of every individual packet within the generated test stream to µs resolution at one or more Observation Points (OPs). Where possible, it is useful to measure the arrival time of the same test stream at multiple observation points within the network, although this is not usually a feature of active test protocols.

TR-452.1 [3] specified the requirements for information capture in §5.1 and for timing in §5.4.

It would also be feasible for an observation point to provide just arrival time metadata about the test stream for analysis rather than any copy of the test stream packets.

5 Using active test protocols for measuring Quality Attenuation

The Session-Sender and the Session-Reflector need to be configured to establish measurement sessions. For measurement protocols that do not have defined control protocols, the measurement session can only be initiated through separate configuration of both the Session-Reflector and the Session-Sender. Then, the Session-Sender starts a performance measurement test and sends test session packets to the Session-Reflector according to the configured packet sending frequency and packet template; the Session-Reflector reflects the test session packets back to the Session-Sender. After receiving the reflected test session packets, the Session-Sender captures the bidirectional information required for ΔQ analysis from the test packets including the sequence number, observed arrival time and any observation timestamps inserted into the test packets by the Session-Reflector. It then calculates ΔQ statistics locally and reports the results to the Controller, or alternatively reports the raw data to a Collator for storage and later analysis. The timestamp information added to the packets by the Session-Reflector allows ΔQ analysis to be performed for both the path from the Session-Sender to the Session-Reflector and the return path.

The following Figure 2 shows a simple ΔQ measurement network topology with a packet reflector.

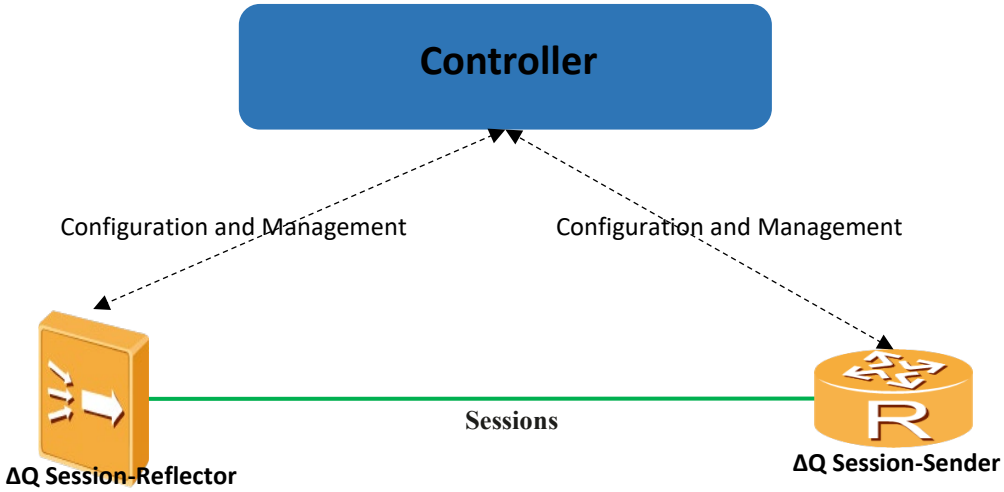

Figure 2: ΔQ measurement network topology

5.1 Test stream generation

In order to retrieve ΔQ measurements, a stream of test packets must be generated and injected into a network path.

[R-1] The Test Stream (abbreviated TS) MUST contain UDP packets that will be routed along the path of interest.

[R-2] The TS MUST contain UDP packets of at least 5 different sizes.

[R-3] The ratio between the smallest and largest packets in the TS MUST be at least 10.

[R-4] The number of packets for each chosen size in TS MUST be configurable, to enable an accurate estimate of the ΔQ of the path of interest. 0 gives guidance as to appropriate values.

[R-5] The TS MUST be compliant to an Active Measurement Protocol (e.g. OWAMP, TWAMP, STAMP, etc.).

5.1.1 Enabling/Disabling Packet Injection

[R-6] The Session-Sender MUST be configurable via a management or control interface.

[R-7] The Session-Sender MUST be able to enable and disable packet injection.

The Session-Sender performs packet injection only after the function is enabled.

5.1.2 Configuring, Executing and Querying Test Sessions

A ∆Q test session is uniquely identified. How it is identified is dependent on the measurement protocols used.

To facilitate management and maintenance of test sessions, the Session-Sender device may further implement a session ID and a session description. The former is used to identify a test session, and the latter is used to describe a test session.
For operational reasons, test sessions in the Session-Sender can be queried. In addition to configuration data, the query result can also contain dynamic data such as the number of packets processed by the test session and the session status

[R-8] A ∆Q test session MUST be uniquely identified. Specifics of the identification method are a subject of the protocol(s) used, as described in 0, ANNEX B, and 0.
[R-9] The Session-Sender MUST support creating and deleting ΔQ test sessions.
[R-10] The Session-Sender MUST be able to execute (start, run and stop) a ΔQ test session at the request of the controller.
[R-11] The Session-Sender MUST be able to support concurrent ΔQ test sessions.
[R-12] The Session-Sender SHOULD support querying of active ΔQ test sessions about configuration, session status and packets count.
[R-13] The Session-Sender MUST be able to detect one-way packet loss.
[R-14] The Session-Sender MUST be able to communicate the ΔQ test results to the Controller or Collator (depending on configuration).
[R-15] The Session-Sender SHOULD have the capability to stream real-time test session results to the Controller /Collator for the duration of the test.

5.2 Test stream reflection

When measuring the ΔQ of a fixed broadband network, the use of Session-Reflector in the measurement process significantly increases the number of potential test endpoints due to the lower implementation complexity of a reflector compared to a generator. Session-Reflector implementation can be further simplified when protocols such as TWAMP light or STAMP (referring to TR-390 [1] and TR-390.2 [2] respectively) are used, because, in these protocols, a Session-Reflector doesn't need a protocol for establishing ΔQ measurement sessions, and there is no need for the Session-Reflector to calculate and report measurement results. The Session-Reflector is simple and only needs to receive test packets, add information, and return the test packets.

This section describes the implementation method of ΔQ measurement and the responsibilities of the device acting as the Session-Reflector. The Session-Sender role is performed by another device or test instrument.

Session-Reflector behaviors are described as follows:

5.2.1 Enabling/Disabling Packet Reflection

[R-16] The Session-Reflector device MUST be configurable via a management or control interface.
[R-17] The Session-Reflector device MUST be able to enable and disable packet reflection.
The Session-Reflector performs packet reflection only after the function is enabled.

5.2.2 Configuring, Reflecting and Querying Test Sessions

A ∆Q test session is uniquely identified. How it is identified is dependent on the measurement protocols used.

For example, for TWAMP and TWAMP lite, according to TR-390 [1], it is uniquely identified by the 4-tuple Source IP, Destination IP, Source UDP, and Destination UDP. A different UDP source port may be used for each concurrent test session.

To facilitate management and maintenance of test sessions, the Session-Reflector device may further implement a session ID and a session description. The former is used to identify a test session, and the latter is used to describe a test session.

For operational reasons, test sessions in the Session-Reflector can be queried. In addition to configuration data, the query result can also contain dynamic data such as the number of packets processed by the test session and the session status.

[R-18] The Session-Reflector device MUST support creating and deleting ΔQ test sessions.
[R-19] A ∆Q test session MUST be uniquely identified. Specifics of the identification method are a subject of the protocol(s) used, as described in 0, ANNEX B, and 0.
[R-20] The Session-Reflector device MAY implement a session ID and a session description.
[R-21] The Session-Reflector device MUST process and return only packets from configured test sessions.
[R-22] The Session-Reflector device SHOULD support querying of ΔQ test sessions about configuration, session status and packets count.
[R-23] The Session-Reflector device MUST enable one-way packet loss detection.

5.2.3 Reflecting Test Packets

The Session-Reflector device receives and parses the test packets sent by the Session-Sender, extracts information from the test packets, constructs test packets based on the extracted information and local information, and sends the packets to the Session-Sender.

The detailed steps are as follows:

1. As soon as possible after receiving a test packet, the Session-Reflector records the time of the packet reception and stores it in the reflected test packet.
2. The Session-Reflector device parses the packets, constructs test packets, and delivers the test packets to the egress interface (see Figure 4). The detail of the format of the packets that are received and transmitted will depend on the specific active test protocol in use. Details for a selection of such protocols are provided in 0, ANNEX B, and 0.
3. The Session-Reflector device stores the transmission timestamp in the reflected test packet just before sending the packet to the Session-Sender device. The timestamp value should be consistently as close as possible to the transmission time of the reflected packet.

[R-24] The Session-Reflector device MUST support sending the test packets back to the Session-Sender device.
[R-25] The Session-Reflector SHOULD implement a mechanism to help the Session-Sender to understand the quality of the timestamps provided.
[R-26] The Session-Reflector MUST transmit symmetric packets to those received from the Session-Sender.

Note: RFC-6038 [13] provides an example of symmetric packets.

5.2.4 Test Session Performance Supported by the Session-Reflector Device

The Session-Reflector device needs to meet certain performance criteria when processing session tests, and must not be affected by the CPU usage.

[R-27] The Session-Reflector device MUST support a minimum (average) interval of 10ms between test packets per test session.
[R-28] The Session-Reflector device MUST support test packet sizes up to 1500 bytes.

5.3 Packet timing capture

Different tools timestamp packets using different methods and at different locations in the end-to-end path. When comparing measurements from different tools it is necessary to be aware of these differences and to be able to quantify them.

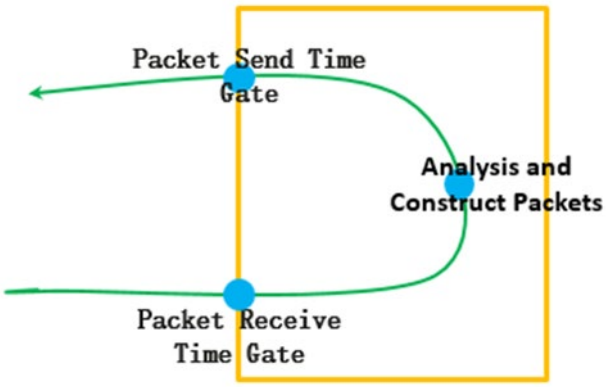

Figure 3: Timing reference diagram

In §5.1 of TR 452.1 [3] R-3 and R-5 defines what metrics are required of a measurement, namely:

- Leading edge time
- Trailing edge time
- And the size

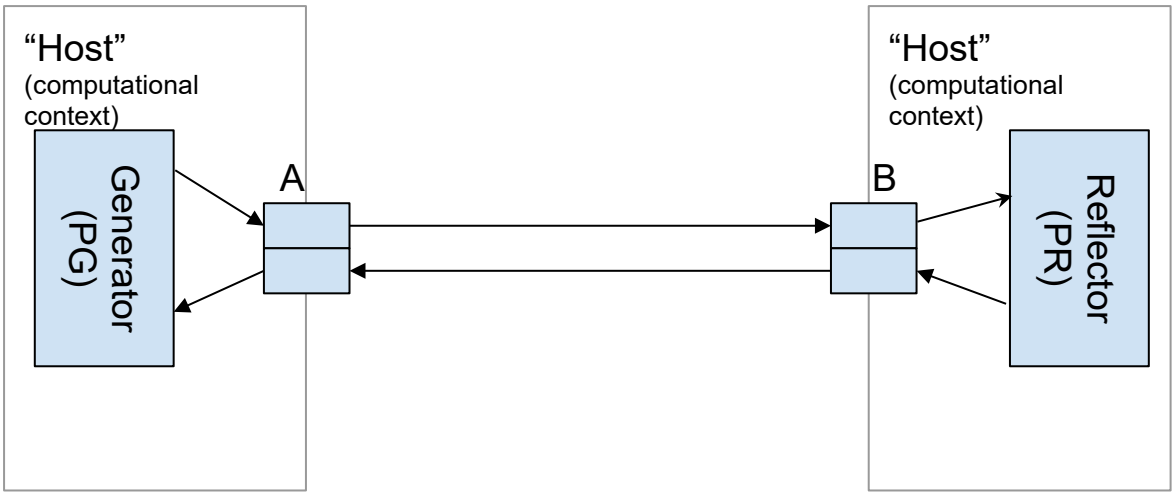

Figure 4: Generic test environment

We are interested in the network latency as seen by higher protocol layers. Consider the generic environment illustrated in Figure 4; any packet traversing from, say, A to B, cannot be processed by a higher protocol layer until it has been completely received. The measure of interest is thus the time of arrival of the last bit at B minus the time of departure of the first bit at A. We call this the 'network ΔQ'; the overall delay (and loss) can only be increased by processing that occurs in the nodes.

Tools like ICMP-ping cannot separate network ΔQ[1] from the ΔQs in the generator host and/or reflector. The reflection ΔQ for ping may be different from the reflection ΔQ of TWAMP or a TCP round-trip measurement, but the network ΔQs for all of these are the same because all the packets in question are transiting the same path.

Ideally, we want to measure the network ΔQs independent of other ΔQs.

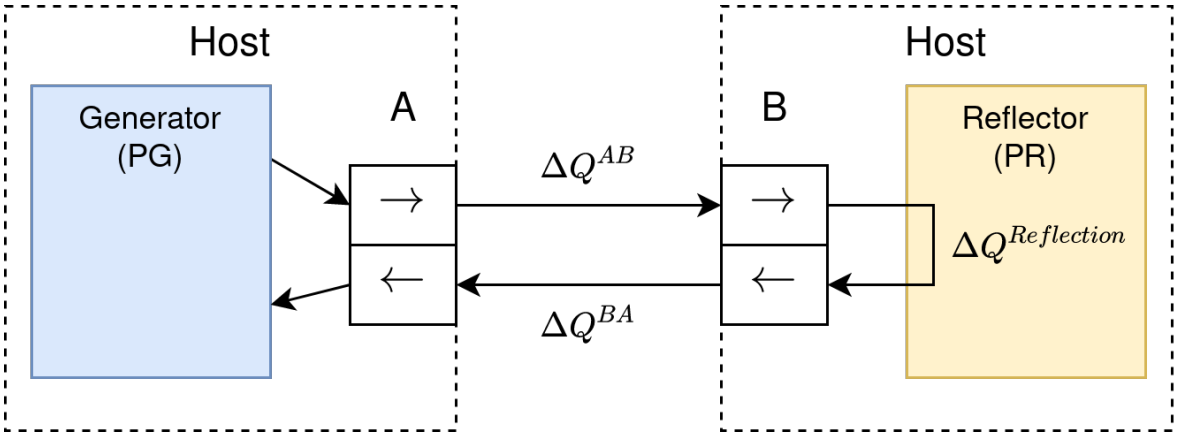

Figure 5: Distinguishing network and reflection ΔQs

As illustrated in Figure 5, we can separate the network ΔQ from the ΔQ of the reflector as long as our measurement process captures the right timestamps[2]. This leads to the following requirements:

[R-29] The Session-Sender MUST record the wall-clock value as close as possible to the start of the transmission of the first octet of the packet.

[R-30] The Session-Reflector MUST record the wall-clock value as close as possible after the last octet of the packet is received.

[R-31] The Session-Reflector MUST record the wall-clock value as close as possible to the start of the transmission of the first octet of the reflected packet.

[R-32] The Session-Reflector MAY provide a more accurate wall-clock value for the sending of the first bit of the reflected packet as a field in a subsequent packet.

[R-33] The Session-Sender and Session-Reflector MAY use means to characterize the accuracy of their wall-clock values.

[1] Remembering that ΔQ incorporates both delay and loss; a packet might be transmitted from A to B but be dropped by the reflector.

[2] Making the metric independent of reflection ΔQ also adds the option of having intermediate observers along the path. This is because, at each observer, the rest of the end-to-end path is just the *ΔQ of the rest of the reflecting path.*

5.4 Observation collation

Observations of packet transmission/reception timings are taken at different points along the network path being measured. In order to compute ΔQ, these observations must be collated together, as illustrated in Figure 6, which implies the existence of a logical function that we call 'collation'. This is related to the Data Collector function defined in TR-304 [6], but with the additional constraint that the data received must be kept in a structured fashion so that ΔQ can be correctly computed. In particular, this means that timestamps for the same packet observed in different locations must be kept together, and the non-observations of a packet (either due to packet loss or observation failure) must be handled appropriately.

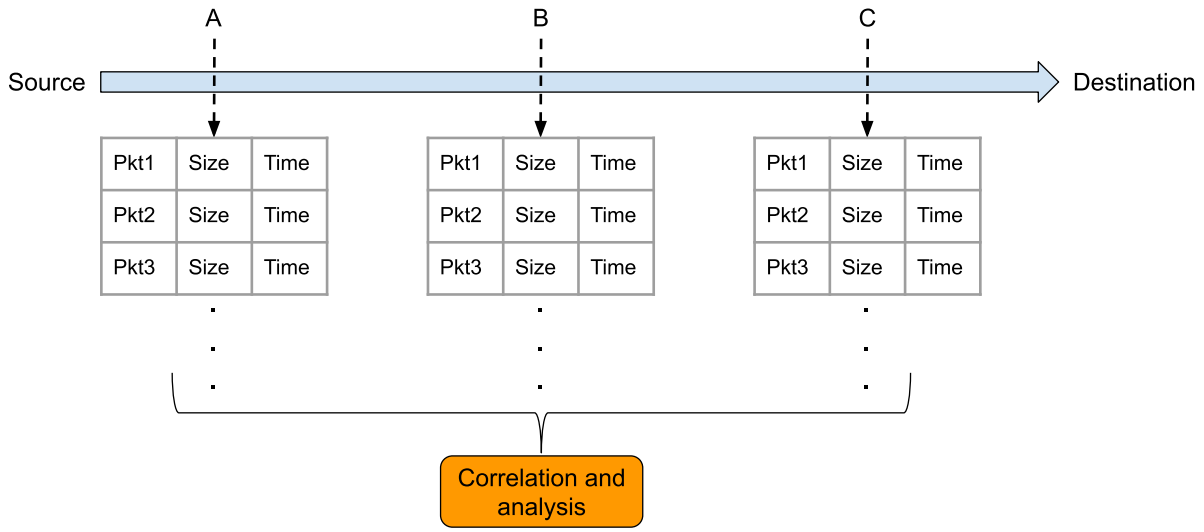

Figure 6 Multipoint observations (From TR-452.1)

In the case of a two-way active measurement protocol where observations are only made by the Session-Sender and Session-Reflector, as shown in Figure 2, this function is already provided by the Session-Sender, as shown in Figure 7.

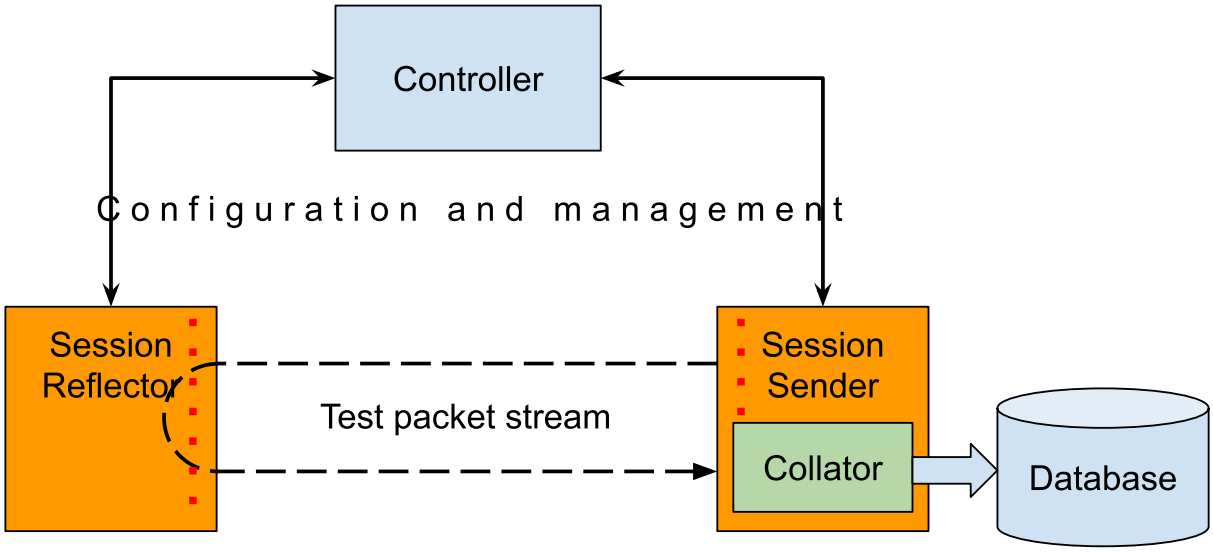

Figure 7 Observation collation

Figure 8 below shows a more complex arrangement including a Collator that combines observations from intermediate points to obtain measures of ΔQ over multiple path segments using only one stream of test packets. If such optional intermediate measurement points are included, their observations cannot be transmitted in-band using fields in the test packets, since these are prescribed by the protocol, therefore they need to be reported by an out-of-band mechanism to the Collator function. The Collator need not be co-located with the Session Sender; in some cases, it may be natural to co-locate collation function with the Controller.

In the case that the active measurement protocol is uni-directional, observations need to be collected from both ends as also shown in Figure 8.

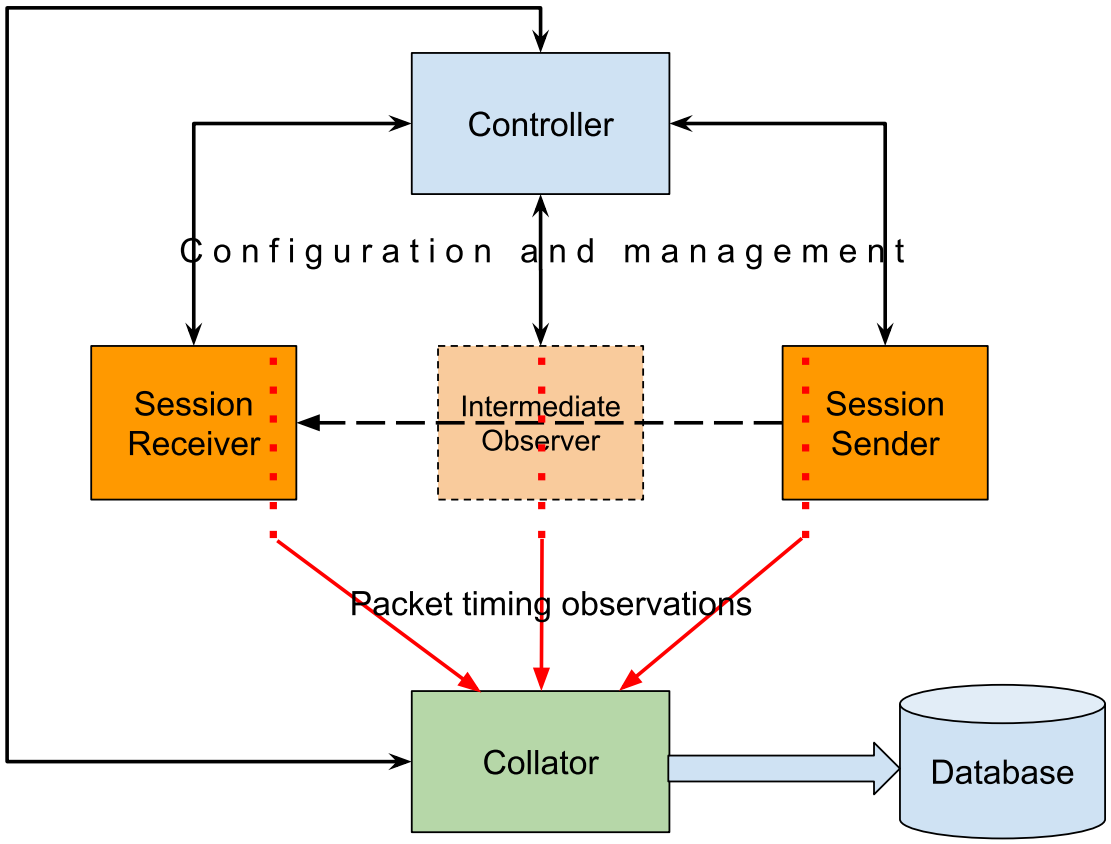

Figure 8 Collation of one-way measurements

Benefits of using a separate Collator function are:

1. The Controller may be responsible for controlling and coordinating a variety of activities, and it may not be appropriate for it to also be involved in the real-time processing of packet timing results;
2. The Collator can translate between the simple protocol to provide results from a Session Sender, Session Reflector or Intermediate Observer to an interface to a database or other analysis function.

Figure 9 below shows how the use of a collator helps with system scalability, by collating results from multiple endpoints.

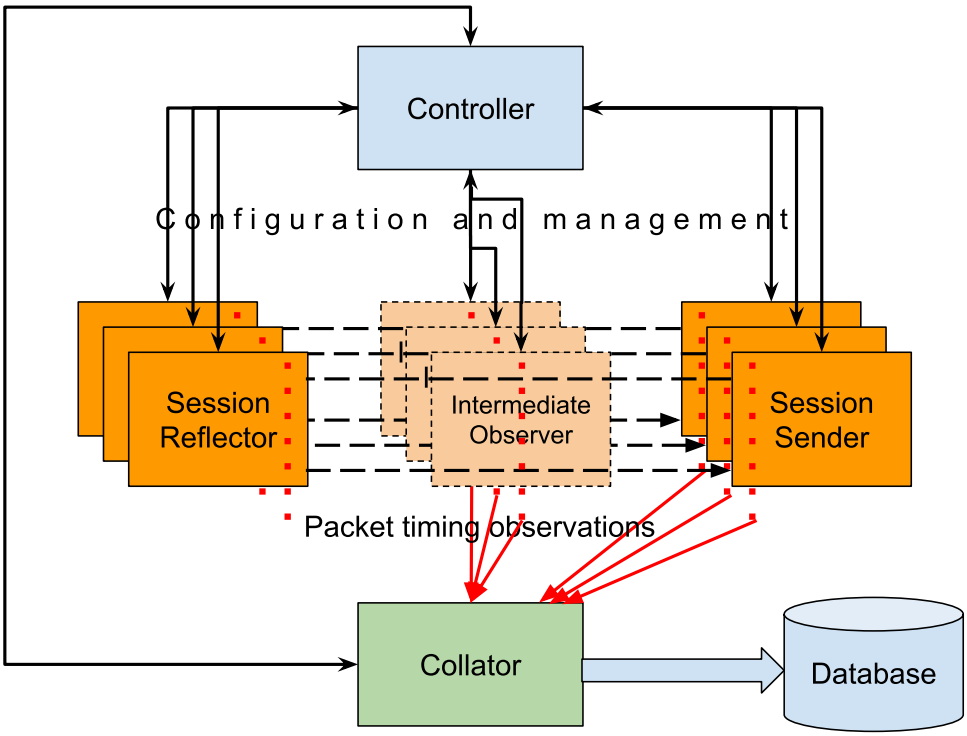

Figure 9 Multiple endpoints communicating with Collator

Figure 10 shows how the scalability can be further enhanced by using multiple Collators (session paths have been omitted for clarity).

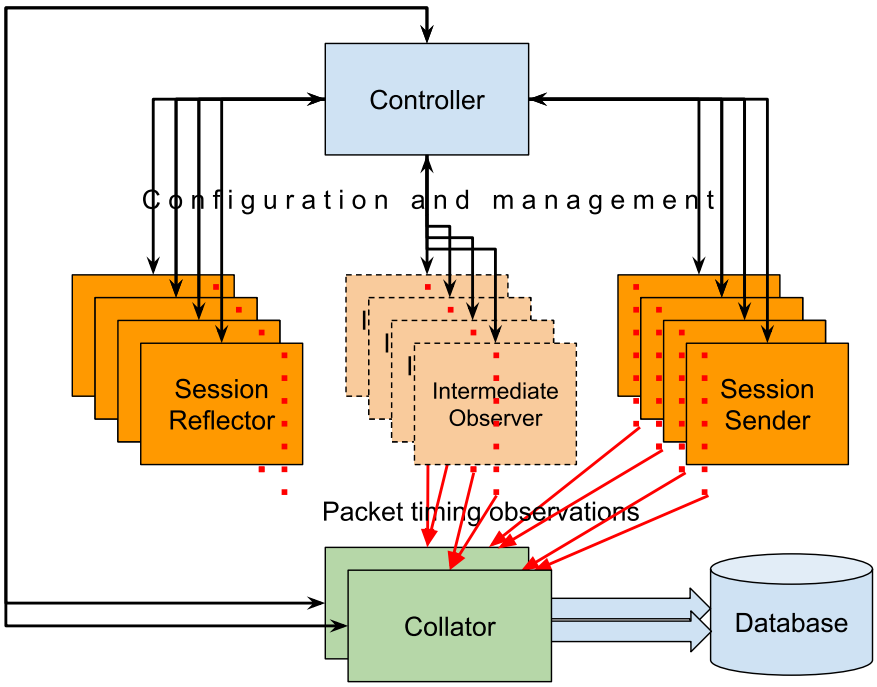

Figure 10 Multiple Collators for higher scalability

[R-34] The Session-Sender MUST report its TS observations to a Collation function. This function may be implicit in the operation of the measurement protocol.

[R-35] The Session-Reflector or Session-Receiver MUST report its TS observations to the same Collation function as the Session-Sender uses. This reporting may be implicit in the operation of the measurement protocol.

If the Collation function is provided by a separate Collator component, the following requirements [R-36] to [R-39] inclusive apply:

[R-36] The Session-Sender and Session-Receiver (if present) MUST be configurable to report their TS observations to a Collator component.

[R-37] If there are any Intermediate-Observers, they MUST be configurable to report their TS observations to a Collator component.

[R-38] TS observations from the same test session made by the Session-Sender and Session-Receiver (if present) and any Intermediate-Observers belonging to MUST be configured to be reported to the same Collator component.

[R-39] Any TS observations reported to a Collator component MUST be associated with a specific test session.

ANNEX A. Specifications particular to OWAMP

One-way Active Measurement Protocol (OWAMP) [12] measures network unidirectional (one-way) characteristics such as delay and loss.

OWAMP, given its specific one-way design, requires a precise time synchronization of the test sender and receiver nodes, therefore it relies on high-precision time sources, such as those provided by GPS.

OWAMP has security features such as an authentication and encryption mechanisms: in this document, for sake of conciseness, just the plain format is shown.

OWAMP consists of two distinct inter-related protocols: OWAMP-Control and OWAMP-Test:

- OWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results.
- OWAMP-Test is used to exchange test packets between two measurement nodes.

The control part of the protocol is beyond the scope of the current document, for further information the reader should refer to the protocol RFC.

OWAMP-Test runs over UDP between a Sender and a Receiver.

The Test Packet Format and Content for unauthenticated mode is shown in Figure 11:

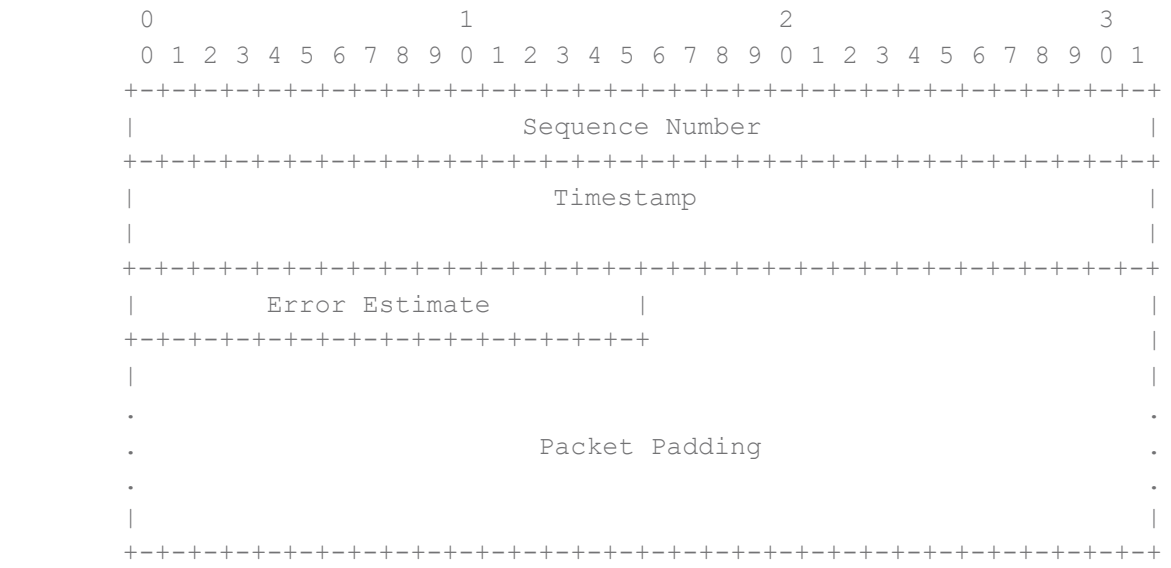

Figure 11 OWAMP-Test Packet Format and Content for unauthenticated mode

The sequence number field starts with zero and it is incremented by one for each sent packet

In the timestamp field (shown in Figure 12), the first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the following 32 bits represent the fractional part of a second that has elapsed since then:

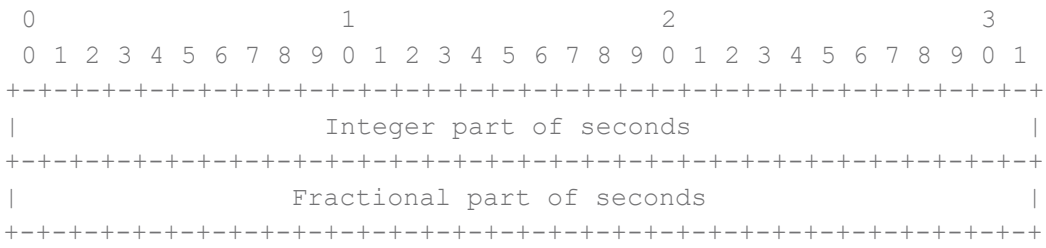

Figure 12 OWAMP-Timestamp field

The Error Estimate field (shown in Figure 13) specifies the Sender time synchronization status and data used to estimate the error:

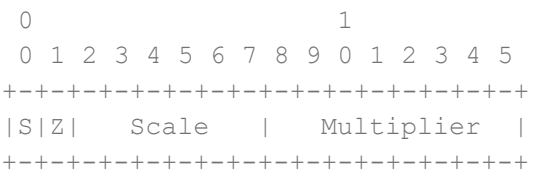

Figure 13 OWAMP-Error estimate field

The Error Estimate field specifies if the time source is externally synchronized (for example with a GPS), plus it gives an estimate of the error in seconds.

ANNEX B. Specifications particular to TWAMP/ TWAMP-Lite within TR-390

Two-Way Active Measurement Protocol (TWAMP) [8] is an OWAMP [12] evolution/extension designed to accommodate two-way measurements (round-trip), without the need of precise time synchronization between the test endpoints.

Any measurement that can be taken using OWAMP [12] can be also obtained using TWAMP.

TWAMP has security features such as an authentication and encryption mechanisms, in this document, for sake of conciseness, just the plain format is shown.

TWAMP consists of two distinct inter-related protocols: TWAMP-Control and TWAMP-Test:

- TWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results.
- TWAMP-Test is used to exchange test packets between two measurement nodes.

The RFC Appendix 1 (TWAMP Light), provides information how to use TWAMP without the need of its Control Protocol. TR-390 addresses the use of TWAMP Light (TWL) for the performance measurement from the IP Edge to the Customer Equipment. TWL was selected since it is a well-defined protocol and its lack of control components makes it lighter requiring fewer resources (CPU, memory, etc.) from the devices that host it.

The control part of the protocol is beyond the scope of the current document, for further information the reader should refer to the protocol RFC [8].

TWAMP-Test runs over UDP to and from a Sender and a Receiver called Reflector.

The Test Packet Format and Content for unauthenticated mode is shown in Figure 14:

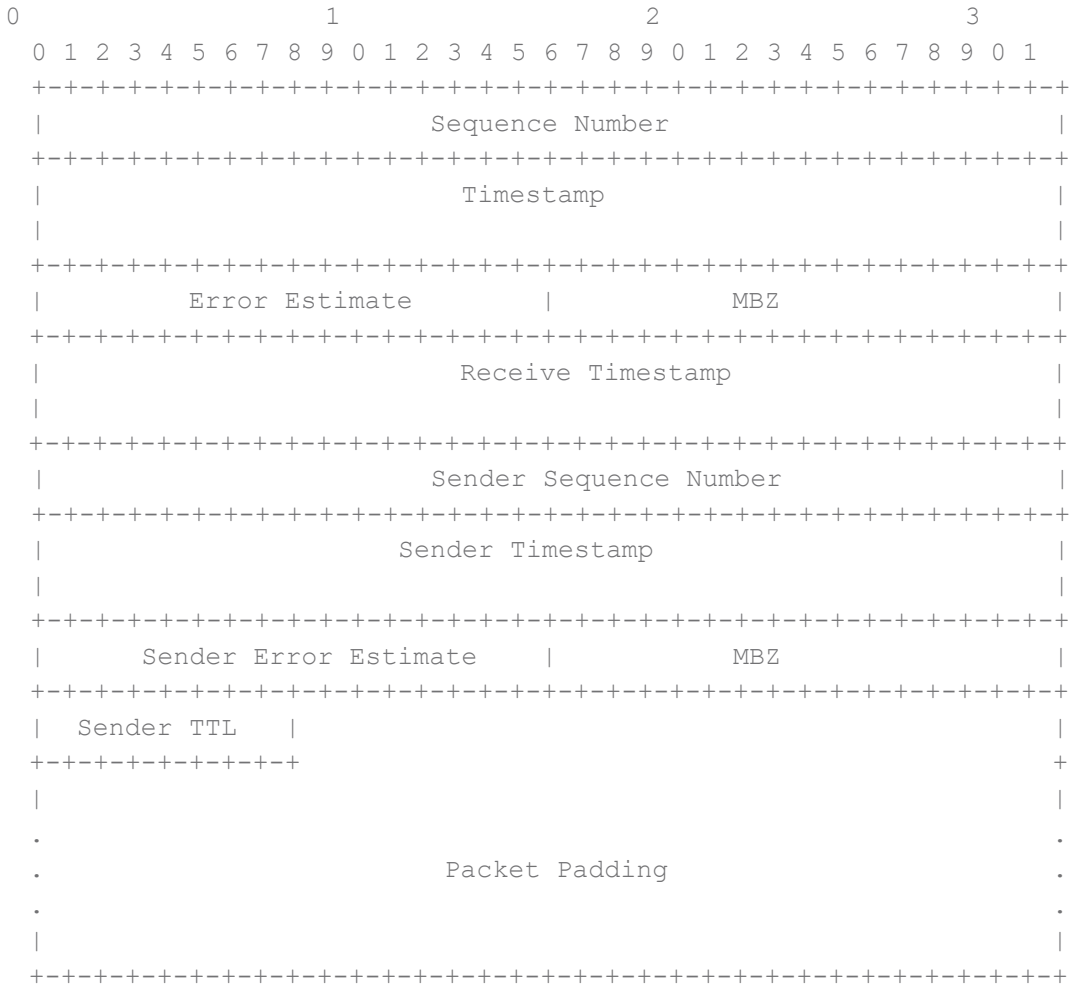

Figure 14 TWAMP-Test Packet Format and Content for unauthenticated mode

Sequence Numbers are the sequence number of the test packets according to their transmit order. It starts with zero and is incremented by one for each subsequent packet. The Sequence Number generated by the Reflector is independent from the sequence number of the arriving packets.

Timestamp is the Sender or Reflector time when the packet was sent, Receive Timestamp is the Reflector time when the packet was received and Sender Timestamp is the exact copy of the Timestamp present in the packet received by the Reflector.

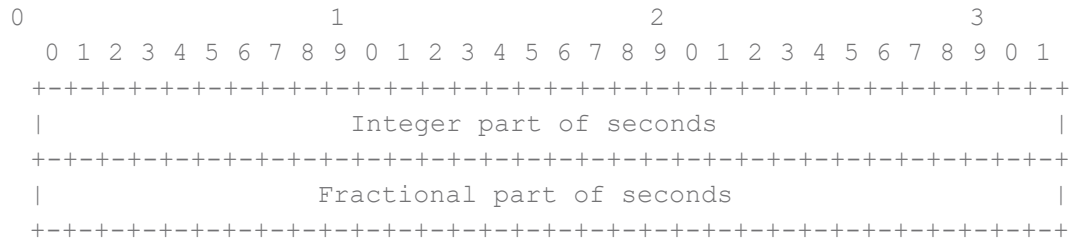

Figure 15 TWAMP-Timestamp field

The Error Estimate field specifies the Sender or Reflector time synchronization status and data used to estimate the error when the packet was sent; while Sender Error Estimate field it's the copy of the Error Estimate field present in the packet received by the Reflector:

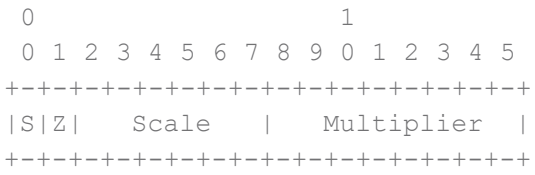

Figure 16 TWAMP-Error estimate field

Sender TTL is set to 255 when transmitted by the Sender. Sender TTL is set to the Time To Live (or Hop Count) value of the received packet from the IP packet header when transmitted by the Reflector:

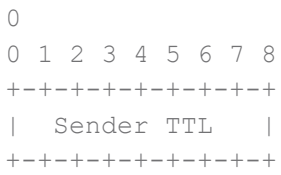

Figure 17 TWAMP-Sender TTL

ANNEX C. Specifications particular to TR-390.2 (STAMP)

Simple Two-Way Active Measurement Protocol (STAMP) [17] enables synthetic packet loss measurement, packet delay, re-ordering, and duplication of packets. Additionally, STAMP extensions [18] support direct loss measurement, generation of STAMP test packets of the variable length, integrity protection, and more. One of the STAMP extensions, the Follow-up Telemetry TLV, can improve the accuracy of packet delay measurement, thus benefiting, for example, measurement of the Quality Attenuation.

The accuracy of the packet delay measurement could be significantly affected by the mechanism used to obtain the wall-clock value at the transmission and reception of a test packet. It is recommended that the wall-clock value is obtained at the same moment of packet transmission and reception. That should minimize measuring delay variance that is introduced by packet queuing. Test systems tightly integrated with a physical network system may achieve consistent timestamping at the cost of portability. A variable delay may get introduced by a function securing the network communication, for example, by calculating the mandatory checksum in the IPv6 network environment or if using an authentication method like the keyed Hashed Message Authentication Code (HMAC). Because the processing can be performed only after a sender of the test packet has updated the timestamp value and the processing time depends on the packet's length, it introduces a variable delay. Hence, separating transportation of the value of the wall-clock when the test packet is transmitted from obtaining that value may improve the accuracy of the delay measurement.

The base STAMP specification [17] allows the traditional method of delay measurement by which three timestamp values are collected in the test packet (Figure 18). The timestamp value at the transmission

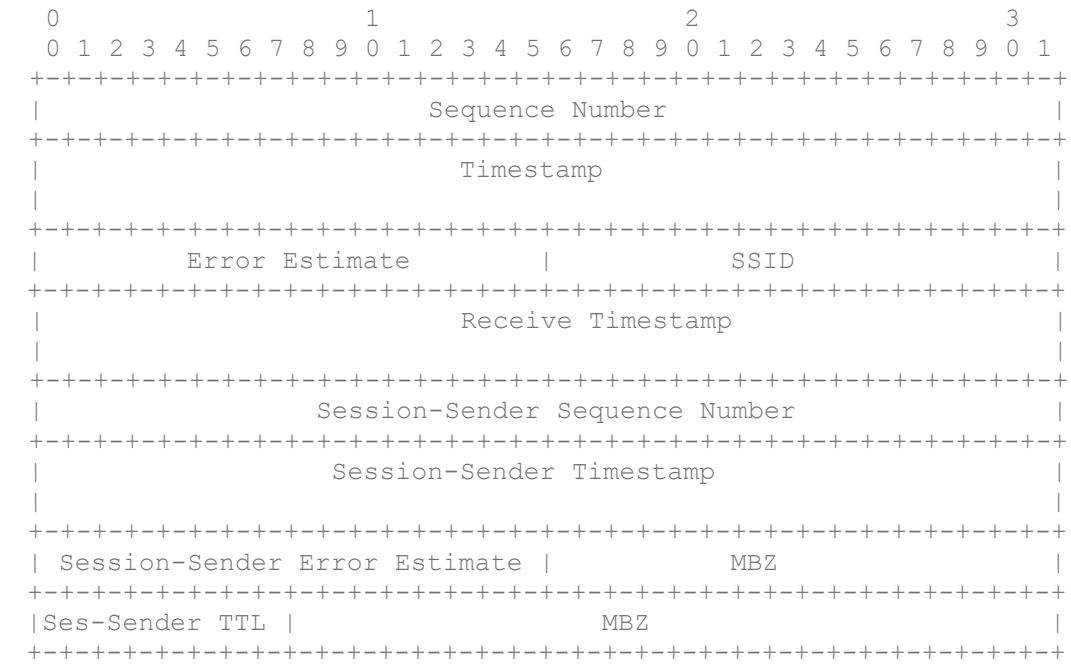

Figure 18 Format of the STAMP Session-Reflector base test packet in the unauthenticated mode

is expected to be directly related to the test packet that carries the value. STAMP functionality is extendable using TLV (Type-Length-Value) construct. Several STAMP extensions are defined in [18]. Among those is the Follow-up Telemetry TLV shown in Figure 19.

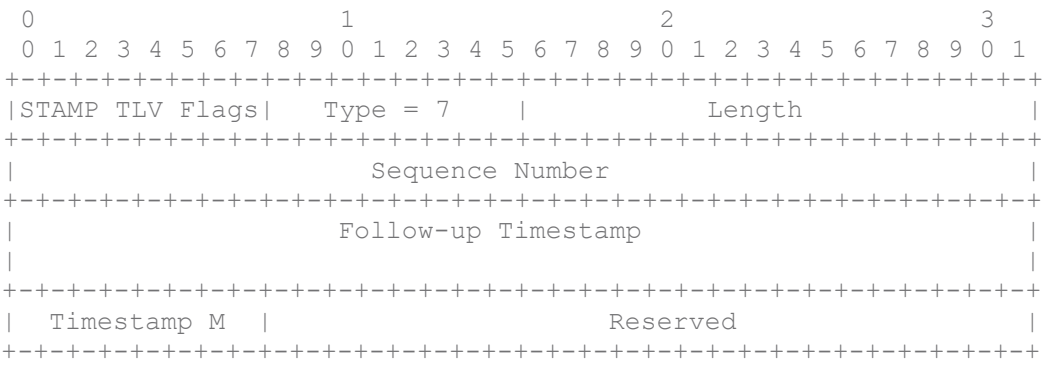

Figure 19 STAMP-Format of the Follow-up Telemetry TLV

The value in the Sequence Number field is the Sequence Number value of the reflected STAMP test packet to which the Timestamp value is applicable. The value in the Timestamp M(ethod) characterizes the method the Session-Reflector used to obtain the Follow-up Timestamp and is one of the values defined in the IANA STAMP Timestamping Methods sub-registry [19]. An implementation of STAMP Session-Sender that supports the Follow-up Telemetry TLV extension includes the TLV in the test packet. The conforming implementation of STAMP Session-Reflector stores the value of the wall-clock associated with the transmission of the reflected test packet and the packet's Sequence Number. These values are copied into corresponding fields of Follow-up Telemetry TLV of the next reflected STAMP test packet. As a result, the Session-Sender receives the reflected STAMP-Test packet with the Follow-up Telemetry TLV that includes the more accurate time at which the preceding packet has been transmitted by the Session-Reflector. The Session-Sender can recalculate delay experienced by that packet and use if for Quality Attenuation.
In addition, the Follow-up extension of STAMP can be combined in a STAMP test packet with the Extra Padding TLV (displayed in Figure 20) to allow the Session-Sender generate test packets of variable length.

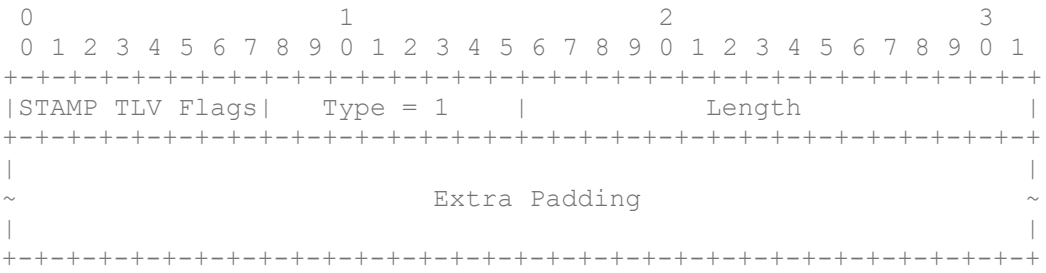

Figure 20 STAMP-Format of Extra Padding TLV

Because the base STAMP specification [17] requires the Session-Reflector by default follow symmetric packet size mode, the conforming implementation will include the same number of padding octets as was in the test packet transmitted by the Session-Sender.

APPENDIX I. Timing accuracy analysis for TWAMP

TWAMP [8] refers to OWAMP [12] §4.1.2 for the details of timestamp representation and content. Although it has error estimates in the packet format, these estimates relate to the local view of NTP accuracy. Nothing appears to be noted as to any systemic infidelities relating what is desired to be measured over what is actually measured. RFC5357 [8] §4.1.1 does encourage "best possible approximation" but without any means of describing the nature of the approximation.

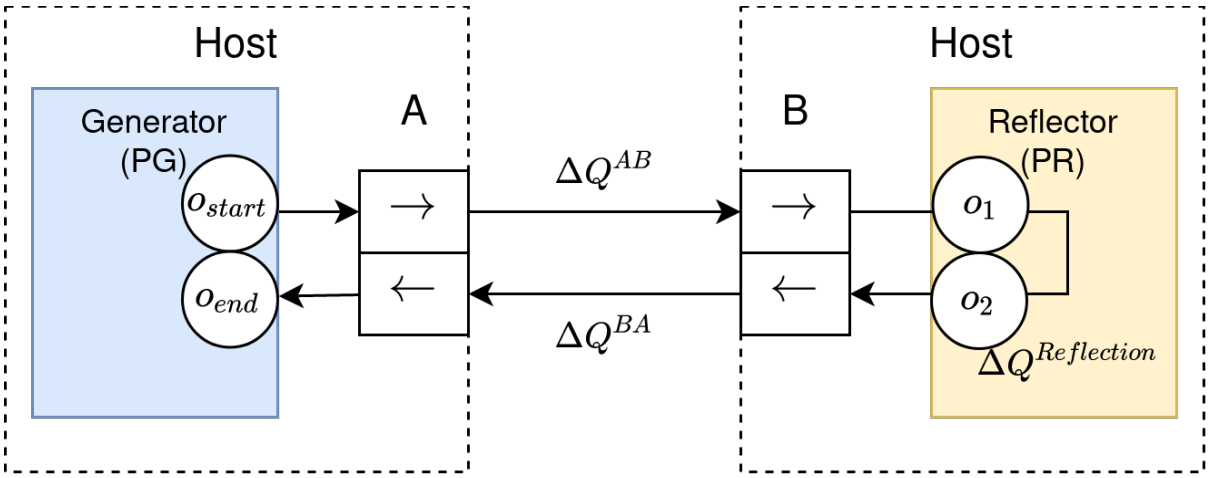

Figure 21: Observation points and ΔQs for TWAMP

Table 1 captures possible/likely infidelities in the realisation of a TWAMP implementation, with reference to the measurement points shown in Figure 21.

TWAMP	QED (ΔQ) observation name			
	$A^{\rightarrow}$	$B^{\rightarrow}$	$B^{\leftarrow}$	$A^{\leftarrow}$
Equivalent Observation	o_{start}	o_1	o_1	o_{end}
Protocol Field	Sender-timestamp	Receive-Timestamp	Timestamp	*not present*
"error"	$\Delta Q^{o_{start}A}$	ΔQ^{Bo_1}	$\Delta Q^{Bo_1} + \Delta Q^{o_1o_2} + \Delta Q^{o_2B}$	$\Delta Q^{Ao_{end}}$

Table 1 Timing references in a TWAMP implementation

APPENDIX II. Relationship of 452.2 to TR-304

TR-304 [6] defines the following terminology:

Access Service Attribute	A parameter that describes a characteristic of a service.
Data Collector	A function that receives Measurement Results reported by a Measurement Agent.
Functional Module	From TR-145 [5]: A set of functions, which can be instantiated in a network node. A network node can contain one or more functional modules. A functional module cannot be split between network nodes. Nodal distribution of functional modules is left to TR-178 [7].
Instruction	The configuration information provided by a Measurement Controller to a Measurement Agent. An Instruction can contain configuration for tasks, schedules and reporting.
Management Server	A function that pre-configures a Measurement Agent.
Measurement Agent (MA)	A function that performs Measurement Tasks under the direction of a Measurement Controller.
Measurement Controller	A function that configures a Measurement Agent.
Measurement Method	A process for measuring the value of a Performance Metric. Where the process involves multiple MAs, each may perform a different role as specified by the Measurement Method.
Measurement Peer	A function that may participate in Measurement Tasks with one or more Measurement Agents. A Measurement Peer does not communicate with a Measurement Controller or a Data Collector.
Measurement Result	A value resulting from the execution of a Measurement Task.
Measurement Schedule	A set of Measurement Task configurations and the times at which they should be performed. The Measurement Schedule is configured in an MA.
Measurement Task	The action performed by a single MA in the determination of the value of a Performance Metric executed at a defined time and with defined parameter values.
Network node	From TR-145: A physical, self-contained element of a broadband network. Examples: a DSLAM, an aggregation switch, etc.
Performance Measurement Framework	Definition of the architecture, functions, and how the functions interwork, to enable performance measurements using standards-based mechanisms.
Performance Metric	From the LMAP framework [14] The quantity related to the performance of the network that we'd like to know the value of.

Report	The set of Measurement Results and associated information that is sent by an MA to a Data Collector.
Reference Point	From TR-145: A reference point is a 'place' inside an architecture, where one or more logical, physical, or business interfaces can be instantiated. A reference point can be internal or can be located at a given physical interface
Report Channel	The address and security information configured in an MA to communicate with a Data Collector
Service Provider	An operator of a data network. This includes providers of Internet service to consumers or businesses, Internet transit services, Content Delivery Networks (CDN), Internet-based applications, as well as enterprise networks.
Suppression	An element in the Instruction that temporarily prevents scheduled measurement-related tasks from being initiated and that may or may not stop currently executing Tasks.

Relating these terms (shown in **bold font**) to those used in this document (some of which are also in TR-452.1), we see:

- Quality Attenuation (ΔQ) is an **Access Service Attribute** and a **Performance Metric**
- The process defined in TR-452.1 is a **Measurement Method**
- Measuring ΔQ over a bounded interval is a **Measurement Task** performed by the Session-Sender; where a Session-receiver or Intermediate Observers are involved, these must also perform **Measurement Task**s
- A Generator is a **Functional Module**
- A Reflector is a **Functional Module**
- A Session-Sender is a **Functional Module** (that includes a Generator) and a **Measurement Agent**
- A Session-Reflector is a **Functional Module** (that includes a Reflector) and a **Measurement Peer**
- A Controller is a **Functional Module** that is a **Measurement Controller**
- A Collator is a **Functional Module** that is a **Data Collector** receiving **Reports**
- Time Gates are **Reference Points**
- A Session-Receiver and an Intermediate Observer are both **Functional Modules** and **Measurement Agents**

APPENDIX III. Number of packets required for statistical accuracy

Recall that the analysis of a series of point to point delays into G, S, and V components involves fitting a line through the minimum delays per packet size. The slope of this line gives the S value and its intersection with the time axis is G. For this to be accurate we need some packets of each size to experience the minimum possible delay. Typically, increased delay results from queuing, so the question is how many packets should we send so that there is a good chance that at least one of them encounters only empty queues. Clearly this depends on both the level of load along the path and the number of queues that are encountered.

We can do a simple analysis assuming that each queue is M/M/1, in which case the chance of it being empty is $1 - \rho$, where $\rho = {}^{\lambda}/_{\mu}$ is the loading factor (assumed < 1). If we pass through *m* such queues and assume that their load is independent, that chance of all of them being empty when we arrive is $(1 - \rho)^m$. The chance that they are *not* all empty is then $1 - (1 - \rho)^m$. If we send *k* packets, the chance that *all* of them fail to find all the queues empty (the situation we wish to avoid) is thus $(1 - (1 - \rho)^m)^k$. If we compare this to a threshold φ, then

$$k \sim \frac{\log \varphi}{\log(1 - (1 - \rho)^m)}$$

We can then compute values of k for different values of ρ and *m*, as shown in Table 2, where $\varphi = 0.05$.

Table 2 Number of packets for $\varphi = 0.05$

	ρ :	0.1	0.2	0.3	0.4	0.5
Number of queues *m*	1	2	2	3	4	5
	2	2	3	5	7	11
	3	3	5	8	13	23
	4	3	6	11	22	47
	5	4	8	17	38	95
	6	4	10	24	63	191
	7	5	13	35	106	382
	8	6	17	51	177	766
	9	7	21	73	296	1533
	10	7	27	105	494	3067

Conversely, for a given value of φ, we can fix *k* and consider what values of ρ and *m* it satisfies. We can do this by observing that

$$\rho = 1 - \sqrt[m]{1 - \sqrt[k]{\varphi}}$$

This is plotted in Figure 22, again for $\varphi = 0.05$.

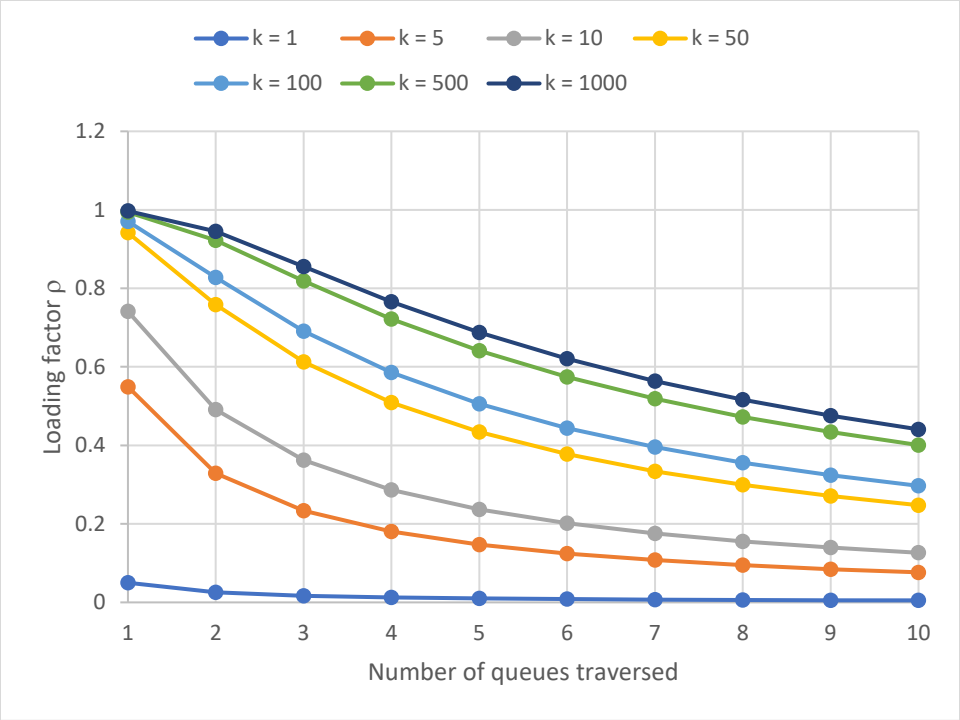

Figure 22: m and ρ as a function of k for φ = 0.05

End of Broadband Forum Working Text TR-452.2