CableLabs
Network Performance

Latency Measurement Metrics and Architecture

CL-TR-LM-Arch-V01-221123

RELEASED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provide any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, infringement, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CL-TR-LM-Arch-V01-221123			
Document Title:	Latency Measurement Metrics and Architecture			
Revision History:	D01 – Released 12/17/21 D02 – Released 4/11/22 V01 – Released 11/23/2022			
Date:	November 23, 2022			
Status:	~~Work in Progress~~	~~Draft~~	**Released**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document designed to guide discussion and generate feedback.
Draft	A document that is considered largely complete but is undergoing review by working groups, members, and vendors. Drafts are susceptible to substantial change during the review process.
Released	A public or gated document that has undergone review. Released guidelines are **not** subject to the Engineering Change process.
Closed	A static document that has been closed to further changes through CableLabs.

Trademarks

Contents

Figures

Tables

1 INTRODUCTION

Low latency is gaining importance in the Internet experience. Low latency is being approached as an end-to-end solution by operators, including Wi-Fi links in the home, with DOCSIS links in the access network and core network segments. Providing lower end-to-end latency is a top priority for operators in the coming years. Measuring the latency in networks, therefore, becomes a vital requirement.

Operators (and third-party speed test websites) have metrics on latency that they have reported and discussed with the community; however, there is confusion surrounding the latency numbers and the ability to compare them between networks. The language and meaning of latency metrics (latency vs. jitter, one-way vs. roundtrip, average vs. 99th percentile), the latency measurement methods, and what is being measured and when (peak vs. off-peak periods) are varied. This report provides clarity around these topics and discusses latency measurement architectures, as well as best-in-class measurement tools to streamline latency measurement for the cable industry.

Operators want the ability to measure the difference in latency that is actually being delivered, before and after they deploy a new technology in their network, like DOCSIS 3.1 AQM, Low Latency DOCSIS®, low latency Wi-Fi, etc. The latency portion of measurement reports (e.g., the FCC's Measuring Broadband America program) are not optimal, and without a consistent measurement approach to latency, this could become a customer perception problem for Internet service providers. For new technologies that differentiate traffic, there are also questions around how latency for unmarked traffic vs. marked traffic can be measured and reported. Operators will be asked to help troubleshoot latency issues, and it will be important for them to identify latency within their networks vs. outside of their networks. This report discusses the latency measurement frameworks that a multiple system operator (MSO) can integrate into its network deployment.

Low latency is gaining importance among operators, and they are focused on reducing latency in each of part of the network, including the Wi-Fi links in the home, with DOCSIS links in the access network and core network segments. Providing lower latency and thus measuring the latency in each portion of network is a vital requirement for MSOs. Operators will need to troubleshoot latency issues and will need the ability to identify latency within their networks vs. outside of their networks.

This report aims to share the experience from developing a simple end-to-end latency measurement framework. A new measurement protocol defined by the IETF is STAMP (Simple Two-Way Active Measurement Protocol, [RFC 8762]). The paper will provide the lessons learned from developing a proof of concept for latency measurement using STAMP. It will describe the high-level measurement architecture and locations for measurement agents and peers. A STAMP reflector could be implemented in a gateway or a device behind it, and a STAMP sender can be implemented somewhere in the network (e.g., in a hub, north of a CMTS). An operator can start with a small number of measurement entities and scale up as needed. If a session-reflector can be dynamically instantiated in a gateway, then one can run measurements on demand. This report will also investigate methods to kick off different latency tests and have the measurement end points report latency data. It will also look into how latency data from various sources can be aggregated and reported. It will also discuss measurement control and reporting methods based on large-scale measurement of broadband performance (LMAP). This report will provide an understanding of MSO needs around latency measurement, an overview of the most appropriate metrics to report, and how to deploy measurement technologies to meet those needs. This report also reports on a prototype STAMP measurement system, which is deployed and collecting latency data.

1.1 Quality of Experience

Latency is the time that it takes for a packet to make it across a network from a sender to a receiver and for the response to come back. Network latency is commonly measured as round-trip time (RTT) and is sometimes referred to as "ping time." As applications turn ever more interactive, network latency plays an increasingly important role for their performance. Applications that are real-time perform the best when latency is low, and adding more bandwidth without addressing latency does not improve the user experience. Packet forwarding latency can have a large impact on the user experience for a variety of network applications. The applications most commonly considered latency sensitive are real-time interactive applications, such as Voice over Internet Protocol (VoIP), video conferencing (such as Zoom), and networked online gaming. However, other applications are also sensitive; for example, web browsing is surprisingly sensitive to latencies on the order of hundreds of milliseconds.

Test results in [G.114] show that highly interactive tasks (e.g., speech, video conferencing, and interactive data applications) can be affected by delays beyond 100 ms, and users report significantly reduced mean opinion scores (MOS) when the voice delays are beyond the 150 ms mark. The current [G.114] recommends a maximum of a 150 ms one-way latency for VoIP applications.

Online games have some models [QoE and Latency] that indicate the impact that network parameters have on user experience. Some data exist to indicate that end-to-end round-trip latency should be kept below 25 ms or 50 ms in order to provide a good user experience, depending on the type of game (first person shooters, massively multiplayer online games, e-sports, etc.). When the operational response delay is less than 50 ms, the MOS tends to be high; when the operation response delay is around 100 ms, the MOS decreases but is acceptable for some kinds of games; and when the operation response delay is beyond 200 ms, the interaction quality for the gamer is very poor.

If we assume that gaming servers centrally located in North America are serving gamers all over the continent, the round-trip time on the fiber backhaul links for gamers on the West Coast will be around 40 ms (assuming 4,000 fiber kilometers between, for example, San Francisco and Chicago, and speed of light in fiber as 0.67× speed of light in vacuum). These RTTs will be even higher for gamers across different continents if they do not have separate gaming servers. Therefore, for the games that require very low latency and latency variance, the 25- to 50-ms end-to-end target implies that the access network latencies need to be consistently in the order of 5- to 10-ms target to meet the requirements for online games.

Web browsing performance is traditionally tracked using page load time. Web content can be sourced from different servers, and web browsers typically fetch resources from each server by opening up multiple TCP connections to the server. As there are multiple handshakes/interactions in each of the underlying protocols (DNS, TCP, TLS, HTTP) and all of those handshakes are impacted by the RTT, higher RTTs increase the page load time. See [Belshe] for experiments on how RTT affects page load time.

1.2 Latency in the Internet

There are a few main contributors to the latency of a packet as it traverses the network. The switching/forwarding delay, propagation delay, and serialization/encoding delay are some of the factors that affect packets as they go across various network devices and links from the source to the destination. Queuing delay is usually the biggest contributor to latency and is mainly caused by the current TCP and its variants. This delay is encountered at the bottleneck links, like home Wi-Fi networks or access networks. The majority of TCP implementations use loss-based congestion control, where TCP ramps up the number of bytes "in-flight" (i.e., its congestion window) until it sees packet loss, cuts its congestion window in half, and then starts ramping back up again until it sees the next packet loss (when the buffers in the device transmit queues are full, a new arriving packet has to be discarded). This way, TCP automatically adjusts its transmission rate to fully utilize the available capacity of the bottleneck link.

The result of this congestion window ramp-up and cut-in-half mechanism is a saw-tooth behavior for the buffer going between partially full and totally full. In every home, there are multiple users and applications that will use the same connection to connect to the Internet. Applications other than TCP will suffer as the packets from those applications will arrive to nearly full buffer that may take tens or hundreds of milliseconds to drain. This can make web browsing perform poorly, and make VoIP, video chat, or online games unusable when other TCP-based applications (e.g., streaming video) are in use.

1.3 Common Techniques to Reduce Latency

Setting the buffer sizes appropriately in each of the network devices is a first step in reducing the latency in the network. Active queue management (AQM) is the next step in mitigating queueing delay, where the basic idea is to detect the increasing queue created by TCP, then drop a packet that will let TCP know to back off on its sending rate, much ahead of the time it takes to drop a packet when the buffer is completely full. There are a variety of algorithms, such as random early detection (RED), Proportional Integral Controller Enhanced (PIE), etc., that an AQM system can implement.

The next stage in the evolution of latency-reducing solutions is the dual-queue approach, where the concept is to separate the traffic for queue-building applications from those applications/traffic flows that are non-queue building. See [White et al.] for detail on these types of traffic flows and the dual queue approaches. Low Latency DOCSIS and Low Latency Low Loss Scalable Throughput (L4S) technologies tackle the queueing delay by allowing non-queue-building applications to avoid waiting behind the full buffers caused by the current TCP or its variants.

2 REFERENCES

2.1 Informative References

[Belshe] "More Bandwidth Doesn't Matter (Much)," M. Belshe, 2010, https://www.belshe.com/2010/05/24/more-bandwidth-doesnt-matter-much/.

[C3 CableLabs] CableLabs Common Code Community, https://code.cablelabs.com.

[CAIDA] CAIDA's active measurement infrastructure, https://www.caida.org/projects/ark/statistics/san-us/med_rtt_vs_dist.html.

[EU Broadband] Broadband Connectivity, European Commission, https://ec.europa.eu/digital-single-market/en/connectivity.

[Excentis ByteBlower] https://www.excentis.com/products/byteblower.

[G.114] ITU-T Recommendation G.114 (05/03), One-way Transmission Time (also Annex B (05/00)), https://www.itu.int/rec/T-REC-G.114.

[Haste] https://www.exitlag.com/en/haste.

[Jitter Calculator] Jitter Calculator, 3rd Echelon Corp., http://www.3rdechelon.net/jittercalc.asp.

[Jones et al.] "NetForecast Design Audit Report of Comcast's Network Performance Measurement System," A. Jones, P. Sevcik, A. Lacy, 2020, https://www.netforecast.com/netforecast-design-audit-report-of-comcasts-network-performance-measurement-system/.

[LM-LULT] Latency Measurement Latency Under Load Tests, CL-AT-LM-LULT-D01-221123, November 23, 2022, Cable Television Laboratories, Inc.

[LM-TRSE] Latency Measurement Test Registry Entries and STAMP Extensions Specification, CL-SP-LM-TRSE-D01-221123, November 23, 2022, Cable Television Laboratories, Inc.

[LM SCTE 20] "Latency Measurement: What is Latency and How Do We Measure It?" K. Sundaresan, G. White, S. Glennon, 2020 SCTE Cable-Tec Expo and Fall Technical Forum, https://www.nctatechnicalpapers.com/Paper/2020/2020-latency-measurement.

[LM SCTE 21] "A Latency Measurement System Using STAMP," K. Sundaresan, 2021 SCTE Cable-Tec Expo and Fall Technical Forum, , https://www.nctatechnicalpapers.com/Paper/2021/2021-a-latency-measurement-system-using-stamp.

[LMAP YANG Tree] IETF LMAP YANG Module, https://yangcatalog.org/yang-search/yang_tree/ietf-lmap-report@2017-08-08.

[MBA FCC] FCC, Tenth Measuring Broadband America Program—Fixed Reports Data and Related Materials, https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-program-fixed.

[MBC CRTC] Government of Canada, Measuring Broadband Canada, 2020, https://crtc.gc.ca/eng/publications/reports/rp200601/rp200601.htm.

[M-Lab NDT] NDT (Network Diagnostic Tool) by Measurement Lab, https://www.measurementlab.net/tests/ndt/.

[Network Next] https://www.networknext.com.

[NQB PHB] IETF draft-ietf-tsvwg-nqb-09, A Non-Queue-Building Per-Hop Behavior (NQB PHB) for Differentiated Services, G. White, T. Fossati, Feb 2022.

[Ozer et al.] "Approaches to Latency Management: Combining Hop-by-Hop and End-to-End Networking," S. Ozer, C. Klatsky, D. Rice, J. Chrostowski, SCTE Cable-Tec Expo 2020, https://www.nctatechnicalpapers.com/Paper/2020/2020-approaches-to-latency-management-combining-hopby-hop-and-end-to-end-networking.

[PingPlotter Pro] PingPlotter Pro Edition tool by Pingman Tools, https://www.pingplotter.com/products/professional; "What is Jitter?" 2016, https://www.pingman.com/kb/article/what-is-jitter-57.html.

[QoE and Latency] "QoE and Latency Issues in Networked Games," J. Saldana, M. Suznjevic, chapter in Handbook of Digital Games and Entertainment Technologies, Springer, 2015, https://doi.org/10.1007/978-981-4560-52-8_23-1.

[RFC 2474] IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, K. Nichols, S. Blake, F. Baker, D. Black, December 1998.

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

[RFC 3550] IETF RFC 3550, RTP: A Transport Protocol for Real-Time Applications, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, July 2003.

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

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

[RFC 6298] IETF RFC 6298, Computing TCP's Retransmission Time, Paxson, V., Allman, M., Chu, J., M. Sargent, June 2011.

[RFC 7223] IETF RFC 7223, A YANG Data Model for Interface Management, M. Bjorklund, May 2014.

[RFC 7317] IETF RFC 7317, A YANG Data Model for System Management, A. Bierman, M. Bjorklund, August 2014.

[RFC 7594] IETF RFC 7594, A Framework for Large-Scale Measurement of Broadband Performance (LMAP), P. Eardley, A. Morton, M. Bagnulo, T. Burbridge, P. Aitken, A. Akhter, September 2015.

[RFC 8126] IETF RFC 8126, Guidelines for Writing an IANA Considerations Section in RFCs, Cotton, M., Leiba, B., and T. Narten, June 2017.

[RFC 8193] IETF RFC 8193, Information Model for Large-Scale Measurement Platforms (LMAPs), T. Burbridge, P. Eardley, M. Bagnulo, J. Schoenwaelder, August 2017.

[RFC 8194] IETF RFC 8194, A YANG Data Model for LMAP Measurement Agents, J. Schoenwaelder, V. Bajpai, August 2017.

[RFC 8545] IETF RFC 8545, Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP), A. Morton, G. Mirsky, March 2019.

[RFC 8762] IETF RFC 8762, Simple Two-Way Active Measurement Protocol, G. Mirsky, G. Jun, H. Nydell, R. Foote, March 2020.

[RFC 8792] IETF RFC 8792, Handling Long Lines in Content of Internet-Drafts and RFCs, K. Watsen, E. Auerswald, A. Farrel, Q. Wu, June 2020.

[RFC 8912] IETF RFC 8912, Initial Performance Metrics Registry Entries, A. Morton, M. Bagnulo, P. Eardley, K. D'Souza, November 2021.

[RFC 8972] IETF RFC 8972, Simple Two-Way Active Measurement Protocol Optional Extensions, G. Mirsky, X. Min, H. Nydell, R. Foote, A. Masputra, E. Ruffini, January 2021.

[SamKnows] SamKnows, https://samknows.com; Quality of Service tests, https://samknows.com/technology/tests/latency-loss-and-jitter#latency-jitter-and-packet-loss-udp.

[Speedtest] Ookla, United States' Mobile and Fixed Broadband Internet Speeds, https://www.speedtest.net/global-index/united-states#fixed.

[TR-452.1] Broadband Forum TR-452.1, Quality Attenuation Measurement Architecture and Requirements, September 2020, https://www.broadband-forum.org/download/TR-452.1.pdf.

[TR-452.2] Broadband Forum TR-452.2, Quality Attenuation Measurements Using Active Test Protocols, November 2022, https://www.broadband-forum.org/technical/download/TR-452.2.pdf.

[White et al.] "Low Latency DOCSIS Overview and Performance Characteristics," G. White, K. Sundaresan, B. Briscoe, SCTE Cable-Tec Expo 2019, https://www.nctatechnicalpapers.com/Paper/2019/2019-low-latency-docsis/download.

[WTFast] https://www.wtfast.com/en/.

[Y.1540] ITU-T Recommendation Y.1540 (12/19), Internet Protocol Data Communication Service - IP Packet Transfer and Availability Performance Parameters, https://www.itu.int/rec/T-REC-Y.1540.

[YANG-HARDWARE] IETF draft-ietf-tsvwg-nqb-08, A YANG Data Model for Hardware Management, A. Bierman, M. Bjorklund, J. Dong, D. Romascanu, March 2018.

2.2 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- IETF: Internet Engineering Task Force Secretariat, c/o Association Management Solutions, LLC (AMS), Fremont, CA 94538; Phone: +1-510-492-4080; Fax: +1-510-492-4001; http://www.ietf.org
- ITU-T Recommendations: Place des Nations, 1211, Geneva 20, Switzerland; Phone: +41-22-730-5111; Fax +41-22-733-7256; http://www.itu.int

3 ABBREVIATIONS

This document uses the following abbreviations.

AQM	active queue management
AWS	Amazon Web Services
bps	bits per second
BSS	business support system
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
CM	cable modem
CMTS	cable modem termination system
CRTC	Canadian Radio-television and Telecommunications Commission
DNS	Domain Name System
DSCP	differentiated services code point
DSL	digital subscriber line
ECN	explicit congestion notification
HMAC	hash-based message authentication codes
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
ISP	Internet service provider
IP	Internet Protocol
ISP	Internet service provider
ITU	International Telecommunication Union
KPI	key performance indicator
L4S	Low Latency Low Loss Scalable Throughput
LMAP	large-scale measurement of broadband performance
LUL	latency under load
MA	Measurement Agent
MBA	Measuring Broadband America
ms	millisecond
MOS	mean opinion score
MP	measurement peer
MSO	multiple system operator
NDT	Network Diagnostic Tool
NOC	network operations center
OSS	operational support system
PDF	probability density function
PDU	protocol distribution unit
PIE	Proportional Integral Controller Enhanced
QoS	quality of service
RED	random early detection
RMS	root mean square
RTT	round-trip time

SDN	software-defined networking
SNMP	Simple Network Management Protocol
STAMP	Simple Two-Way Active Measurement Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TLV	type-length-value
TOS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol

4 VIEW OF LATENCY MEASUREMENTS

Internet latency is crucial in providing reliable and efficient broadband services to customers who are connecting to servers across the country and the globe. The trend of real-time gaming and other real-time applications only accelerates the importance of accurately understanding the latency characteristics of the network. This bubbles up the task of latency measurement toward the top of an operator's priority list. Being able to accurately diagnose latency issues seen by residential or business customers is becoming more important. In order to support server selection in distributed/virtual computing environments, measuring accurate latency becomes extremely important. Knowing the latency characteristics well allows an operator to make better decisions about which latency-reducing technologies to deploy and where.

Accurately measuring network latency, however, is not an easy task due to lack of testing end points, lack of clock synchronization when needed, the sheer volume of collected data points, and aggregating and analyzing the data meaningfully. In addition, the time that latency is measured affects measurement results significantly due to network dynamics, volatile traffic conditions, and network failures.

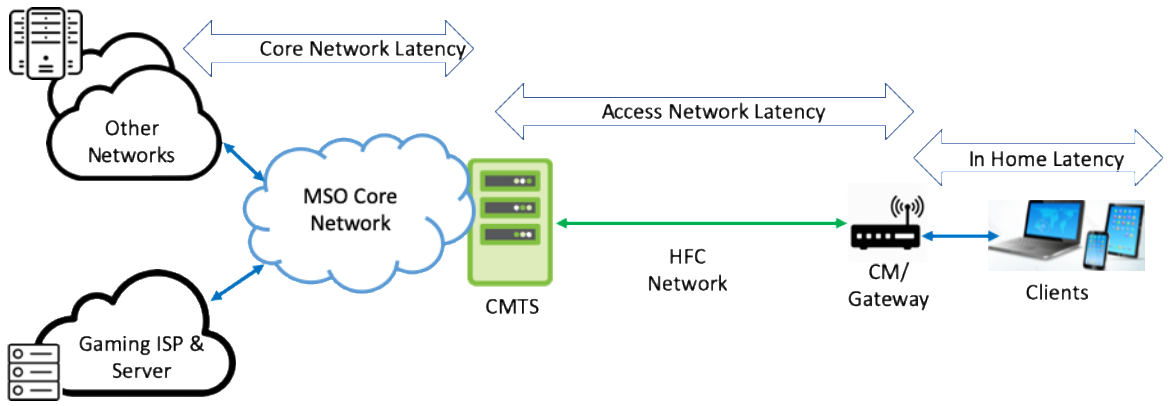

Figure 1 - MSO View of Latency Measurements

4.1 MSO Goals for Latency Measurement

Operators want to leverage existing available tools and standardized architectures to quickly set up a measurement infrastructure. Some of the common operator use cases and considerations are as follows.

- Identify latency in third-party networks vs. MSO core network vs. home network: In the core network, there is a need to develop processes to identify routing issues, especially in the path to the egress point in the network, which connects to a specific application server. For the access and home networks, it is extremely useful for an operator to be able to delineate latency from within the customer's home (because of Wi-Fi) vs. the access network latency vs. the aggregation/core network.
- Operation diagnostics support: Operators would like to develop diagnostic tools so that they can give meaningful information to their operations teams. The use of latency measurements in network operations center (NOC) and field tester tools for live problem diagnoses is common at IP and Ethernet layers.
- Operators would need to measure a variety of access and core architectures (R-PHY, FMA, Integrated) and need the measurement methods to work across this range of deployments.
- Network architecture analysis: The loss and delay performance metrics impact the scalability of the network and also its behavior under load. For network architects, understanding both end-to-end network latency plus the contribution of the various links and nodes (network devices) that the network comprises is very useful.
- Understanding how to optimize the network deployments: With distributed CCAP architectures, an operator has to decide on a particular architecture or where to place the physical or virtual components and decide on the location of certain functionality (e.g., MAC scheduler).

- There are many benchmarking purposes for which the latency measurement data can be used; for example, different equipment (switches, routers) introduce different degrees of delay when processing packets. When moving from physical network elements to virtualized network elements, an operator needs to be able to quantify the latency difference.
- Lab latency measurements can compare the impact of introducing a new network element or configuration (e.g., a new technology like Low Latency DOCSIS) and verify the end-user experience prior to deployment.
- Optimizing network configuration: Appropriate latency measurement techniques can help diagnose intermittent issues (e.g., buffer overflows) and help fix them.
- Now with a goal of identifying per-hop latency, operators need to identify the appropriate locations for the measurement end points: end device, gateway/cable modem (CM), CMTS, router, interconnection point, etc.
- Any measurement architecture needs to support frictionless deployment of latency measurement infrastructure. This is dependent on how the specific measurement infrastructure is implemented and deployed (e.g., is it using hardware probes vs. virtual probes?). Scalability of the measurement platform across an entire operator becomes an important consideration.

4.2 Latency Measurement and Reporting by Third Parties

Broadband infrastructure is gaining the attention of various national communications regulators as countries focus on enabling their people with high-speed Internet connectivity. As a part of this, many of these regulators measure the broadband deployments and report on various metrics, such as houses covered, speed tiers available, etc., and also conduct network measurements on actual upload and download speeds. Latency measurements are also becoming an integral part of these reports.

4.2.1 Measuring Broadband America

In the United States, the FCC runs the Measuring Broadband America (MBA) program. The MBA program is a nationwide study of consumer broadband performance, and it collects network performance data from a representative sample of customers from each of the fixed Internet service providers (ISPs) (see [MBA FCC] for the latest speed and latency data reported). The MBA program tests that are conducted are automated, direct measurements of customers' service during a single month and are done in collaboration with the measurement company, SamKnows. Each volunteer customer connects a "Whitebox" device to their home network that performs the tests after finding the nearest test servers.

The MBA program measures latency by measuring the average round-trip time from the consumer's home to the two closest measurement servers, one server chosen from each of two "pools" of servers. The report shows the median latency for each participating ISP and includes aggregated information for each ISP and type of access network. It reports the measured latencies for various DSL, cable, and fiber-based ISPs on an individual basis, as well as aggregated. The MBA program has a limited number of test server locations in each pool. Only six cities host test servers in both pools (an additional four cities host a server in only one pool). This means that client devices that are geographically distant from these six cities will report latency numbers that are more likely to be correlated to geography than to network capability. Difference in geographical distance to the server and the distance of the number of hops internal and external to the ISP network can make a difference in the number of network links the test packets have to travel across and, ultimately, the latency measured.

The MBA program latency and packet loss tests measure the round-trip times for approximately 2,000 packets per hour, sent at randomly distributed intervals. Per [MBA FCC], the latency and packet loss test records the number of packets sent each hour, the average round-trip time, and the total number of packets lost (a packet is considered lost if the packet's round-trip latency exceeds 3 seconds). The test computes the summarized minimum, maximum, and standard deviation and mean from the lowest 99% of results. MBA determines the mean value over all the measurements for each individual's Whitebox and then computes a median value from the set of mean values for all the Whiteboxes.

4.2.2 Measuring Broadband Canada Project

The Canadian Radio-television and Telecommunications Commission (CRTC) has commissioned a study of the performance of broadband services sold to Canadian consumers. This project measures broadband Internet

performance, including actual connection speeds, in Canadian homes. The CRTC collaborated with a number of Canadian Internet service providers and SamKnows and produced a Measuring Broadband Canada Report in June 2020 (see [MBC CRTC]). The report describes that, unlike in the U.S. MBA program, the latency data was focused on Whiteboxes located within a 150-km radius of the test server locations in order to minimize the effect of distance on measurements. See [MBC CRTC] to understand the details on the average latency during peak hours for different Canadian service providers and access networks (cable, DSL, fiber). Like the MBA report, [MBC CRTC] also measured packet loss and average webpage loading times from a selection of websites.

4.2.3 EU Broadband Report

The European Commission has a vision for broadband connectivity and takes policy actions to turn Europe into a "Gigabit Society" by 2025. In support of this, the European Commission has commissioned a study to obtain reliable and accurate statistics of broadband performance across the different EU member states and other countries.

4.2.4 Speedtest by Ookla

Speedtest by Ookla publishes the [Speedtest] market reports as a guide to the state of fixed broadband and mobile networks around the world. Each report mainly includes speed data (downstream and upstream) and insights about country trends. Speed test data is based on the results of millions of tests run by Speedtest users. An individual user-initiated speed test uses "ping" to report the latency to the nearest Speedtest server. Speedtest is very relevant in the latency measurement landscape, as that is how the majority of consumers understand what their service speed is and what latencies their connections achieve. Of course, consumers also tend to run Speedtests when they see an issue with the service or when they upgrade or get a new service, so this may also not be a representative sample across consumers.

5 LATENCY METRICS

Each operator needs to track different metrics, or key performance indicators (KPIs), when it comes to network latency. The network latency metrics important to operations teams will be different than what metrics are important to product or regulatory teams. Metrics can also be dependent on where the network is in the product lifecycle. There are a variety of latency metrics to choose from, and this section describes how to look at and understand latency.

As a packet travels across a network, the packet experiences different types of delays at intermediate hosts, routers, and network links. A host or router needs time (processing/switching/forwarding delay) to process an incoming packet to determine its next hop. The packet also often waits in the transmit queue behind other packets (queuing delay). Transmission delay (serialization/encoding delay) is the time for a node to move out all the bits of the packet onto the link. Finally, it takes time for the packet to propagate over the link from one node to another. End-to-end latency is the sum of such delays at every step of the way.

5.1 One-Way Latency (or Packet Delay)

One-way latency is the total time it takes for a packet of data to travel from the sender to the receiver across one or multiple hops. The one-way delay will be dependent on congestion of the network at the time the packet was sent. It will also depend on the topology of the network and the distance and routing decisions between the two end points. Measuring one-way latency also implies that the sender and receiver have synchronized clocks, which is sometimes a challenge to set up and maintain when the end points are across multiple network domains.

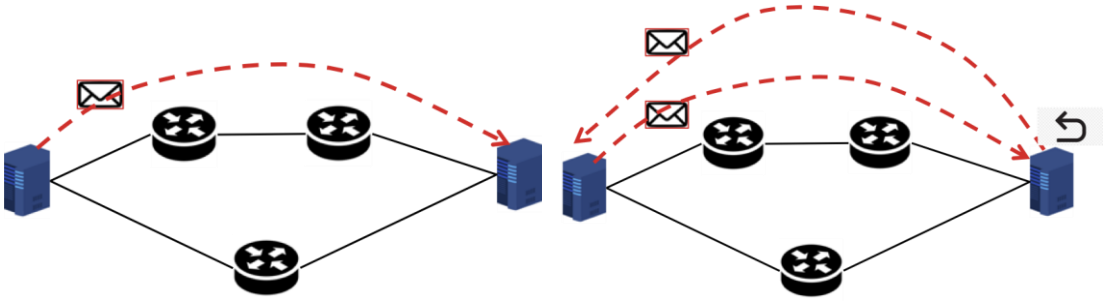

Figure 2 - One-Way Latency vs. Round-Trip Latency

5.2 Round-Trip Latency

Round-trip time (or round-trip latency) is the time taken for a packet of information to travel from the sender to the receiver and back again. RTT is the total time it takes for a packet of data to travel from the sender to the receiver, across one or multiple hops, plus the total length of time it takes for the receiver to send the packet back to the sender, through one or multiple hops.

The round-trip latency is more often quoted, as it can be measured from a single point. It requires a process running on the other end to mirror the packet back. The RTT can vary if the return path is different from the forward path. The most common example for round-trip measurements is the Internet Control Message Protocol (ICMP) Echo Request/Reply, used by the ping tool.

5.3 Singleton Measurements vs. Sets of Measurements

A singleton measurement test can send one packet and calculate the one-way or round-trip latency of that packet. That sample is not the most interesting, as it is just one sample on a network carrying millions of packets. Latency varies with different factors, such as the location of the two measurement end points, and with time (because of changes in route selection or congestion). Therefore, most latency measurement tests use multiple packets in a test for one-way or round-trip measurements. This gives an operator a sample distribution of latency measurements and paints a better picture of the latency behavior. A test would measure the latency of each of the test packets, allowing

an operator to understand what the behavior of the network latency is across that set of packets. Having more data samples allows the operator to observe the variations and better understand the network latency in a way that correlates to what the customer will perceive and experience.

Operators typically run each of these tests multiple times a day to get a feel for the network latency variation over time. These sets of measurements could be performed over time for one user or across multiple users, or it could be both—tests done over time and for multiple users.

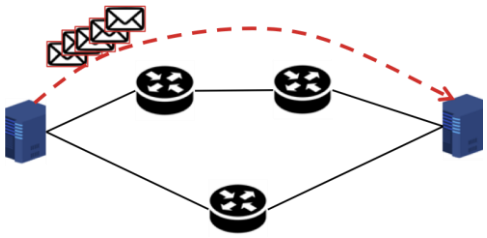

Figure 3 - Sets of Latency Measurements

5.4 Jitter or Delay Variation

The term "jitter" is commonly used to refer to variation in the latency of arriving packets over time. Even though it is prevalent in networking parlance, the term is considered deprecated by technical bodies like the IETF. The IETF [RFC 5481] now uses the term "delay variation" for metrics that quantify a path's ability to transfer packets with consistent delay. Note that "jitter" can also be used to convey undesired variation in signals in contexts beyond IP packet transfer (e.g., frequency or phase variations in electronic circuits in reference to a clock or sampling jitter in analog-to-digital conversion of signals).

The term "jitter" can be defined within a specific context in order to provide a meaningful metric for a specific application, and it is sometimes defined simply in a manner that is convenient to calculate. Most real-time voice and video applications employ a de-jitter buffer to smooth out delay variation encountered on the path. Many of the commonly used jitter definitions are aimed at helping designers of such systems choose the size of the de-jitter buffer.

[RFC 5481] notes that various standards for delay variation have allowed flexibility to formulate the metric, and so the specific formulations of delay variation must be well understood. All definitions of delay variation are derived from the one-way or round-trip delay metrics. The networking industry has predominantly implemented two specific formulations of delay variation: inter-packet delay variation (IPDV) and packet delay variation (PDV).

5.4.1 Inter-Packet Delay Variation

A latency test or application will send a sequence of packets to measure one-way or round-trip latencies. Inter-packet delay variation is derived from such a sequence of latency measurements. It is simply the difference in latency of each packet as compared to the previous packet.

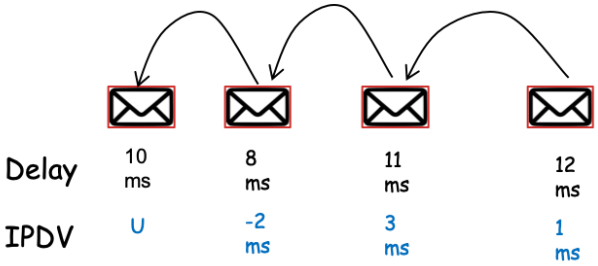

Figure 4 - IPDV Calculation

IPDV is sometimes used when an in-band, real-time metric is needed, such as for continuous monitoring of an isochronous protocol, since it can be calculated on the fly (and in the isochronous case, it can be calculated simply using the receive timestamp of each packet). Because this calculation is inherently influenced by the packet rate, it is less useful for active measurements (see Section 6.1.1); it could be difficult in general to align the packet rate used for measurement with the packet rates used for the various user applications that may be of interest.

Furthermore, the IPDV calculation is inherently a first-order high-pass filter applied to the packet latency time series, and thus it amplifies rapid changes (e.g. packet to packet) in latency while being extremely insensitive to slower variations.

5.4.2 Packet Delay Variation

Packet delay variation is also derived from a sequence of latency measurements where a single reference latency is chosen from the stream based on specific criteria. The most common criterion for the reference is the packet with the minimum delay in the sample. Other references, such as average latency, can also be chosen. PDV is simply the difference in latency of each packet as compared to the one reference packet. In [Y.1540], the ITU also chooses this definition of packet delay variation.

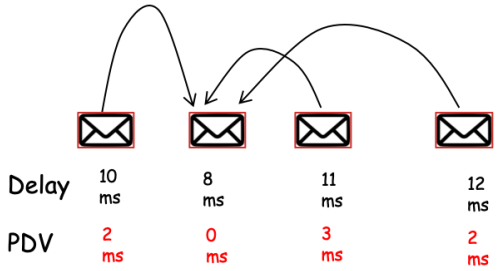

Figure 5 - PDV Calculation

5.4.3 Jitter Metrics in Use in the Industry

The formulations described in the previous sections result in a per-packet metric, which (across a set of packets) can then be summarized using descriptive statistics (mean, median, standard deviation, median absolute deviation, P99, P99.9, etc.) in order to come up with a summary of the delay variation across the set of packets.

There are different ways in which jitter definitions are used in different applications in the industry. [SamKnows], [Haste], [Excentis ByteBlower], [M-Lab NDT], and [Network Next] use statistics derived from the PDV definition, whereas [WTFast], [Jitter Calculator], [RFC 3550], and [PingPlotter Pro] use statistics derived from the IPDV definition. Table 1 describes the definitions each of these entities use and how they aggregate it. There are significant differences in the meaning of a term from one entity to another.

Table 1 - Jitter Definitions in the Industry

Entity	Parameter	Definition
PDV-Based		
[SamKnows] *(Network performance measurement platform)*	Jitter	P99 PDV (PDV referenced against min latency)
[Excentis ByteBlower] *(Byteblower traffic generator)*	Jitter	Standard deviation of PDV
[Haste] *(Optimized routing for game traffic)*	Jitter	Standard deviation of PDV
[Network Next] *(Optimized routing for online games)*	Jitter	jitter = 3* RMS(PDV) (PDV referenced against min latency)
[M-Lab NDT] *(Network test)*	Jitter (round-trip time variation)	max(PDV) (PDV referenced against min latency)

Entity	Parameter	Definition
IPDV-Based		
[RFC 3550] *(RTP Protocol)*	Interarrival jitter	Exponentially weighted moving average of the absolute value of IPDV
[PingPlotter Pro] *(Ping statistics tool)*	Jitter	Average of the absolute value of IPDV
[Jitter Calculator] *(Internet Services company)*	Jitter	Average of the absolute value of IPDV
[WTFast] *(Gaming VPN solutions)*	Jitter	Average of the absolute value of IPDV

5.5 Packet Loss

Packet loss is another metric to quantify an IP network's ability to transfer packets in both directions from one host to another host; failure to transfer a packet in either direction constitutes a round-trip packet loss. Packet loss may be indicative of network congestion, especially at peak usage times. Packet loss also maybe caused by presence of noise on the link, leading to loss of the transmitted signal.

In a latency measurement test, packet loss is typically reported as a fraction of the test packets lost during a latency test.

In many cases, network protocols (including latency measurement protocols) will not wait indefinitely for a particular packet to arrive and will instead set an upper bound on this wait time, with packets arriving after this deadline treated identically to packets that were dropped in the network.

Traditionally, TCP has used 3 seconds as the initial retransmission timeout (RTO). [RFC 6298] recommends lowering this RTO value to 1 second. Based on this, many latency measurement tests will declare a packet to be lost in a round-trip test if a response is not received within 3 seconds.

When making the choice of timeout threshold for packet loss, one should be aware that some protocols/application may discard excessively delayed packets. Even though the measurement system might not report those beyond the application's threshold as packet loss, as a network operator one may need to consider those higher latencies for debugging applications.

Packet loss in latency measurement is typically handled in one of two ways, either by separately counting packet loss events during the measurement (and excluding these events from any descriptive statistics that are generated based on the remaining packets) or by treating a lost packet as having infinite latency. Either approach can be valid depending on the context.

Some of the descriptive statistics and visualizations discussed in the next sections are ill-defined when lost packets are treated as having infinite latency, but others (such as a CDF plot) can handle this without issue.

5.6 Descriptive Statistics

Once it has received a set of measurements (each of which is an individual latency measurement), a network operator wants to easily aggregate and make sense of those sets of measurements across the whole network and over time. The question is how best an operator can analyze the data to guarantee that the latency meets service requirements.

5.6.1 Basic Statistics

Many operators start with basic statistics, like mean, median, or min-max. Each of these numbers have their place, but for large populations of data, they often hide the actual network behavior. Mean and median tend to hide outliers, especially the high-latency events, which may happen only during specific times. In contrast, the maximum is overly conservative and is easily distorted by a single outlier event.

- Average—The arithmetic mean, or average, is simply the sum of the set of the latency measurements divided by the number of measurements. The set of results of each experiment or an observational study can yield its own average number. For latency measurements, even though the average may be a starting point, it hides a lot of the variation in latency. Some of the much higher excursions are diluted by the mean,

and, thus, averages hide high-latency events, which would ultimately impact the customer experience. Outliers also skew averages, so the average does not represent typical behavior either.

- Min/Max—The maximum and minimum of a set of measurements are the largest and smallest value in the set of measurements. These are useful to understand the limits of the network. In the context of latency measurements, one can separate lost packets as a separate measure instead of considering it as infinite latency.
- Standard deviation—The standard deviation is a measure of how spread out the latency measurement numbers are. The standard deviation is calculated as the square root of the variance (average of the squared differences from the mean) for each sample. This gives a measure of the amount of variation or dispersion of a set of latency values. A low standard deviation indicates that the values tend to be close to the mean of the set, whereas a high standard deviation indicates that the values are spread out over a wider range.

5.6.2 Percentile Numbers

Many descriptive statistics, like mean, standard deviation, and skew, are most meaningful when the underlying data follows a roughly normal (Gaussian) distribution. In contrast, even simple latency distributions are often heavily skewed with a set of values around a certain range and with many fluctuations and outliers. As a result, these traditional statistics offer very little value in capturing or describing latency, but percentiles can generally be much more effective.

Percentiles allow a better understanding of latency distributions than averages. A percentile is a value below which are a certain percentage of observations. Percentiles show the point at which a certain percentage of observed values occur. For example, the 95th percentile is the value that is just greater than 95% of the observed values, i.e., 95% of packets got a lower latency than the P95 value. For example, to obtain the 99th percentile of a collection of latency measurements from a network, an operator can sort them and discard the highest 1% of values. The largest remaining value is the 99th percentile. This value is the largest latency that will be seen for 99% of the measurements. An operator can choose a measure like the 90th, 95th, 99.9th (or even more nines) percentiles, which are typically denoted as P90, P95, P99, etc.

Network latencies between machines can be low when the network path is idle, but when there is significant network activity, packets can take anywhere from a few milliseconds to hundreds of milliseconds, or even seconds. Because many network segments (particularly broadband links) are idle, or nearly so, for a significant portion of the day, the median latency and the minimum latency are often pretty close to one another. Long-tail latencies occur when the higher percentiles begin to have values that are many times greater than the median. In a long-tail latency distribution, the 99th percentile can be 50 times greater than the median value, far beyond normal distributions.

Percentiles are often used to find outliers. When a range of percentiles is computed, one can estimate the data distribution more accurately. Another use for latency percentiles is to implement a threshold beyond which issues are flagged to the operator. An operator could also track a combination of a few different percentiles, such as P50, P75, P95, and P99, and flag issues when any of them change significantly with respect to previous measurements or thresholds.

The next step is to determine which latency response time metric is more representative of the user experience (e.g., the 95th percentile or the 99.9th percentile). Table 2 shows how to think about the impact to an application like gaming. Gaming traffic flows are typically 60 packets per second at a rate of 100–200 kbps in the upstream direction and 60 packets per second at a rate of 500 kbps–1 Mbps on the downstream. Gaming clients or servers expect packets to arrive at that consistent rate of 60 times per second, and any packets that arrive with a much higher latency cannot be used and are essentially thrown away. As an example, 99% of the gaming packets have a latency of 40 ms or less, and 1% of packets are delayed for anywhere from 100 ms to 500 ms. For a real-time game, this 1% "latency event" happens (on average) once every 1.6 seconds, and such network behavior is unwelcome in the gaming environment and may be a showstopper in other applications. Based on this view, perhaps the P99.9 value would be a good starting point to represent user experience for online gaming.

Table 2 - Understanding Latency Percentiles

Notation	Percentile Latency	Meaning	Implication	Impact for a Gaming Application
P50	50th percentile—median latency	50% of packets got this latency or better	50 of 100 of packets got worse than this latency	Every other packet!
P90	90th percentile	90% of packets got this latency or better	10 of 100 packets got worse than this latency	6 packets per second
P95	95th percentile	95% of packets got this latency or better	5 out of 100 packets got worse than this latency	3 packets per second
P99	99th percentile	99% of packets got this latency or better	1 of 100 packets got worse than this latency	1 packet every 1.6 seconds
P99.9	99.9th percentile	99.9% of packets got this latency or better	1 of 1,000 packets got worse than this latency	1 packet every 16.6 seconds
P99.99	99.99th percentile	99.99% of packets got this latency or better	1 of 10,000 packets got worse than this latency	1 packet every 2 mins 46 seconds

Recommendation: This working group has chosen to track the following set of 10 values corresponding to the 0^{th}, 10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th}, 95^{th}, 99^{th}, 99.9^{th}, 100^{th} percentile.

5.7 Histograms

A histogram is a graphical method for displaying the shape of a distribution and is particularly useful when there is a large number of observations. To construct a histogram, the range of values observed in the measurement is divided into a series of intervals, or bins. The measurements are then classified into bins counting how many values fall into each interval. The bins usually are specified as consecutive, non-overlapping intervals of a variable. The bins/intervals are contiguous and are often of equal size.

The goal is to collect enough data points for good latency characterization. This means an operator needs to collect data to obtain acceptable precision for different percentile levels. A simple process in latency measurement is to record all the latency data over multiple sets of tests and then later analyze and sort the data into traditional histograms to get the required percentile data. Some alternatives to the traditional histograms with linear bins are logarithmic bins or arbitrary bins. Linear bins in histograms require a large amount of storage to cover the range with good resolution, and logarithmic bins cover a wide range of values but do not have good precision. Arbitrary bins work only when the operator already has a good feel for the interesting parts of the data range.

5.8 Visualization of Latency

Data visualization can reveal patterns and trends in the data, allows quick absorption of large amounts of data by network operators, and ultimately lets the operator understand the information and make decisions. This section describes some of the ways in which an operator can visualize latency data.

5.8.1 Time Series

A time series is a set of observations (x_t) ordered in time and observed at a discrete set of (approximately) evenly spaced time intervals: at times t =1,2,…, N , where N is the length of the time series. The figure below, created using [PingPlotter Pro], shows a time series of ping data, once every second for 10 minutes. The average ping time is ~12 ms; however, that is not the normal case, and there are many latencies of 15–20 ms and occasionally even up to 25–30 ms.

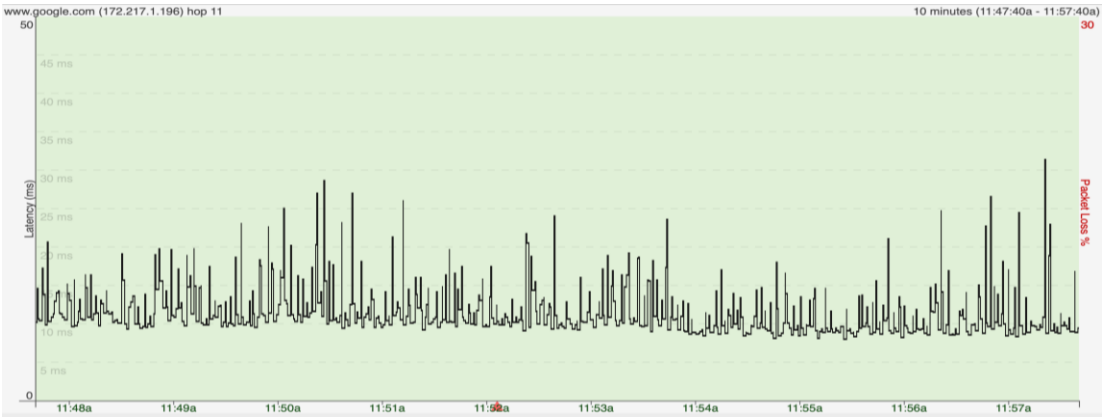

Figure 6 - Example Time Series of Latency Measurement

5.8.2 Probability Density Function (PDF)

A probability density function is a statistical expression used in probability theory as a way of representing the range of possible values of a continuous random variable. For a continuous function, the probability density function is the probability that the variable has the value *x*. The area under the curve represents the probability that the variable will fall within an interval, and it is expressed in terms of an integral between two points ($Pr[a \leq X \leq b] = a\int^{b} fX(x)dx$).

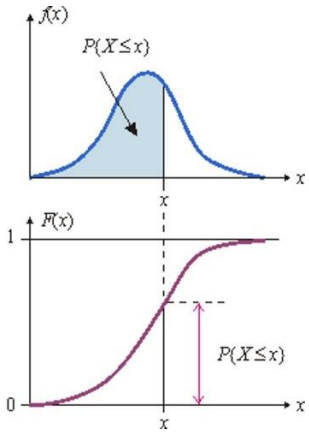

Figure 7 - PDF–CDF Relationship

5.8.3 Cumulative Distribution Function (CDF)

The cumulative distribution function of a random variable is another method to describe the distribution of random variables. A cumulative distribution function describes probability that a random variable takes on a value less than or equal to *x* ($Pr[X \leq x] = FX(x)$). Though it is common to see CDF plots that span the full range from 0 to 1, as shown in Figure 7, note that if packet loss events are treated as having infinite latency, the amount of packet loss in a particular measurement manifests itself as the gap between the highest CDF value and 1.

5.8.4 Complementary Cumulative Distribution Function (CCDF)

A complementary cumulative distribution function answers the opposite question, i.e., how often the random variable is above a particular level *x*. To obtain the cumulative distribution function, the integral of the PDF is computed, then the CDF results are inverted in the CCDF. (The CCDF is the complement of the CDF, or CCDF = 1 – CDF.) The CCDF can also be plotted in a logarithmic scale so that the more interesting percentile values are easily discernible.

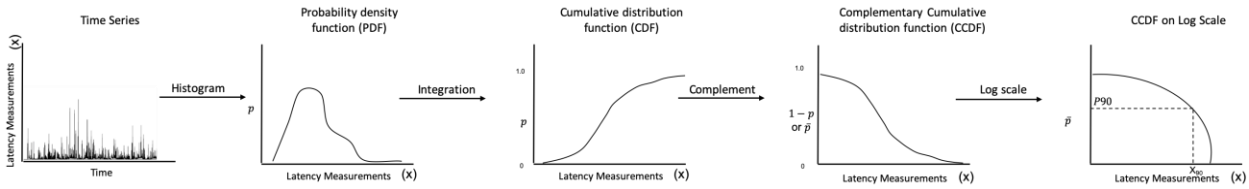

Figure 8 - Conversion from Time Series to PDF to CDF to Logarithmic–CCDF

5.8.5 Example of PDF/CDF/CCDF

Figure 9 shows an example of latency measurements performed in the lab of round-trip times from a client to a server, which are separated by a Wi-Fi link and a DOCSIS link (CM and CMTS) with a pseudo Low Latency DOCSIS configuration. The latency measurements of unmarked traffic are shown in blue, and the latency of differentiated services code point (DSCP)-marked traffic is shown in orange. Figure 9 also shows the various latency and jitter metrics of the unmarked traffic and how varied the numbers can be.

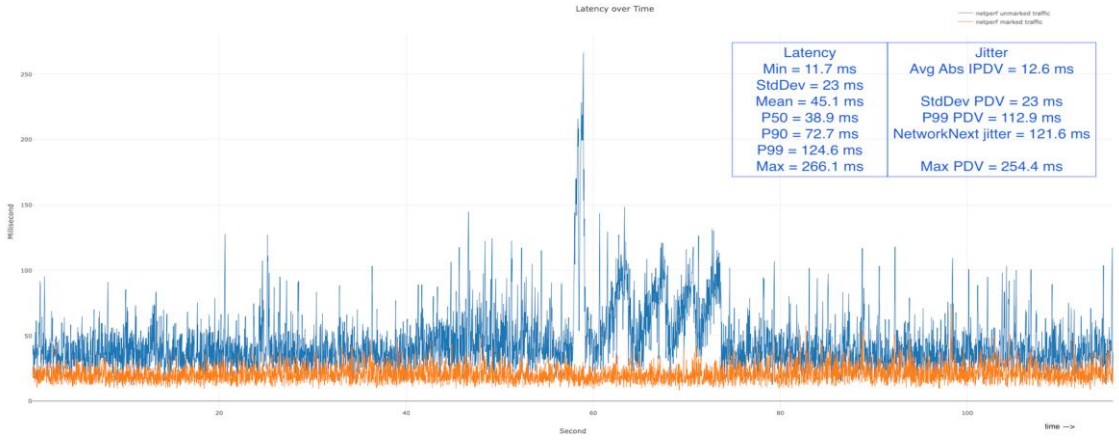

Figure 9 - Time Series Latency Data of Marked vs. Unmarked Traffic

Figure 10 shows how different the two sets of latency measurements are using a histogram of 1-ms bins, with the marked traffic flow (orange) having lower and tighter latency numbers and the unmarked traffic (blue) having latencies that extend from 100 ms all the way to 260 ms.

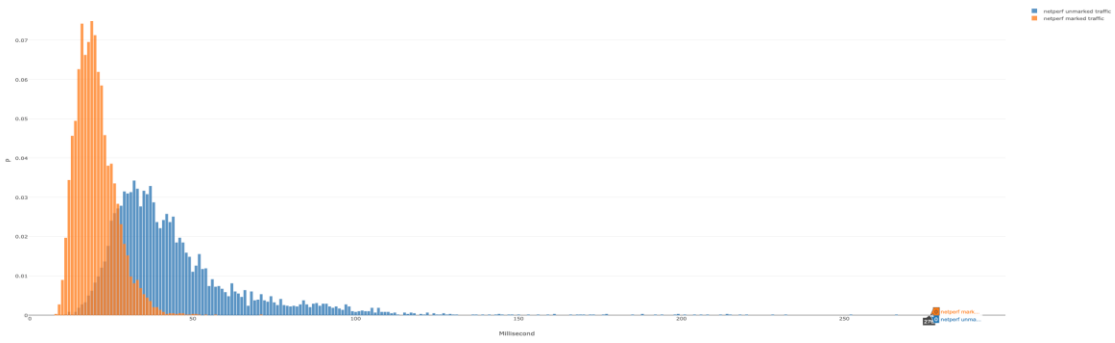

Figure 10 - Probability Distribution Function (PDF)

Figure 11 shows the cumulative distribution function for the same dataset, with the marked traffic flow (orange) having a lower P99 (~38 ms) and the unmarked traffic having a higher P99 (~125 ms).

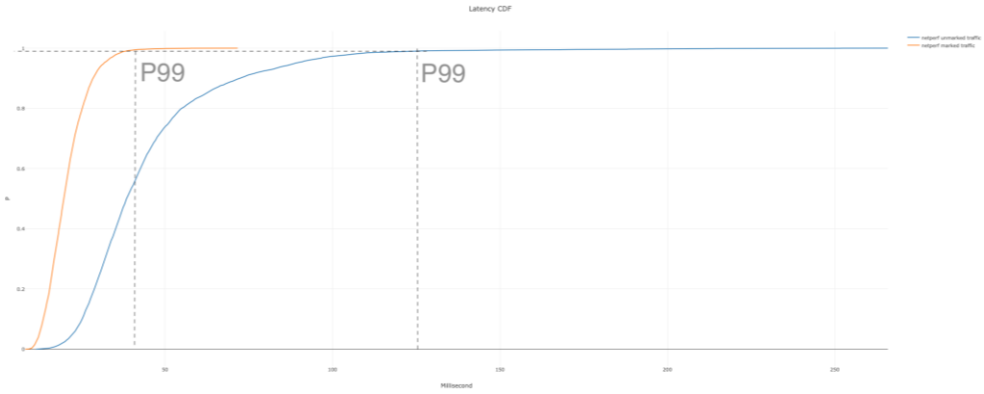

Figure 11 - Cumulative Distribution Function (CDF)

Figure 12 shows the complementary cumulative distribution function for the same dataset. It is essentially the same graph but inverted, with P99 readings closer to the bottom of the graph compared to the top.

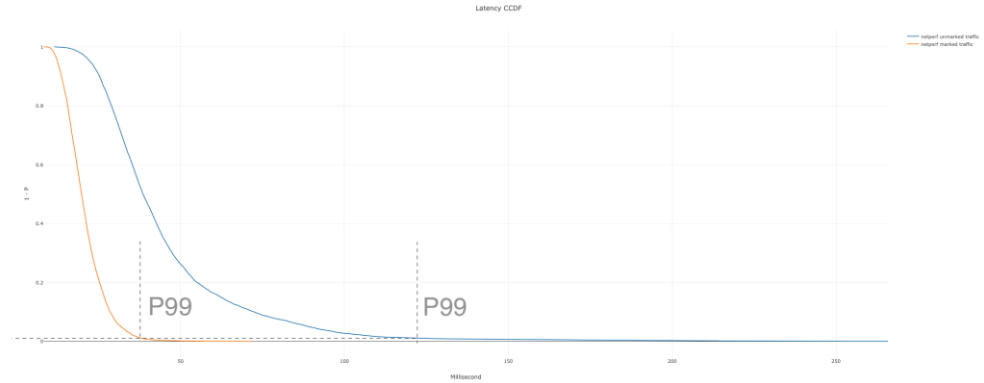

Figure 12 - Complementary Cumulative Distribution Function (CCDF)

Figure 13 is the same CCDF graph but is drawn on a logarithmic scale for both axes. The P90, P99, or P99.9 can be compared, and the differences can be seen clearly at this scale in the percentiles that we are interested in.

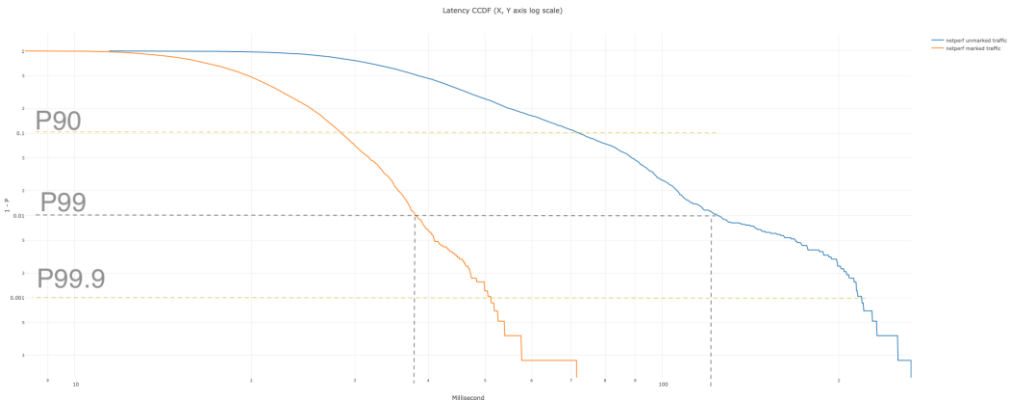

Figure 13 - CCDF on a Logarithmic Scale

6 LATENCY MEASUREMENT APPROACHES

6.1 Types of Measurement

6.1.1 Active Measurements

Active measurements are conducted by generating traffic between two end points for the sole purpose of measuring the latency. For example, with ICMP ping, a sender sends an ICMP packet(s) to the receiver, who replies back; the sender calculates the time between sending and receiving the packet(s). The measurement is considered to be active, as the reason the traffic is created and sent is to measure the latency between the end points.

Active monitoring involves injecting test traffic into the network, typically with the same forwarding criteria as the user traffic being monitored, then measuring its performance. These tests can be either one-way (from site "A" to site "D" or round trip (from site "A" to site "D" and back to site "A") depending on what the operator wants to measure. Because the test traffic mimics the user traffic, active testing gives a packet-by-packet view of the end-to-end performance of a network with regard to such things as latency, delay variation, or packet loss.

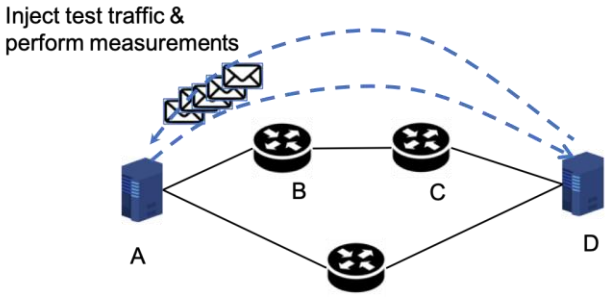

Figure 14 - Active Measurements

Active testing can be performed between successively longer paths along the network route, for example, from site "A" to site "B" or site "A" to site "C." During this testing, the operator can segment the overall end-to-end path so that performance indicators can be derived on a per-segment basis, which shows where issues possibly are located. Active monitoring is the primary method for policing service-level agreements because it provides a real-time view of performance. Active monitoring requires two end points to be able to create test traffic and respond back to complete the measurement.

6.1.2 Passive Measurements

Passive measurements are done simply by observing normal host-host interactions. Instead of measuring the latency of specially created test packets like in active measurements, passive measurements are based on the normal user packets that traverse the network. Passive measurements observe the traffic exchanged between two end points and calculate the latency based on observed activity. For example, during normal interactions between host A and D, like during the initial handshake, a packet sent from A to D would be immediately responded to by D as per the normal protocol interaction. If this transaction can be observed, for example, at a location B, one can measure the time between sending the packet and receiving the response. Passive methods obtain similar measurements to active measurements without creating any new test traffic in the network but are reliant on the presence of user traffic and, therefore, can be skewed (for better or for worse) toward periods of time when more such traffic is present. Passive monitoring involves capturing and analyzing live network traffic, or traffic statistics, at a specific point in the network; for example, at the network interface to an application server or at an aggregation router.

Passive monitoring does not require another host in the network to be involved in the process. Passive monitoring involves capturing some, or all, of the traffic flowing through a port for detailed, offline analysis of things like signaling protocols, application usage, or top bandwidth consumers. Passive monitoring is suited for in-depth traffic and protocol analysis and can give visibility into the customer's actual quality of experience.

6.1.2.1 TCP Analysis

Analyzing the delay experienced by the TCP connection setup packets is an example of passive measurements. TCP uses a three-way handshake to establish a reliable connection. The TCP connection setup consists of a handshake with SYN, SYN+ACK, and ACK packets. The idea is to examine the data for outgoing connections and compute the round-trip delay between the SYN & SYN+ACK packets, as well as the SYN+ACK & ACK packet in the handshake. Because TCP connection end points normally respond immediately, this is an easy way to compute the round-trip times.

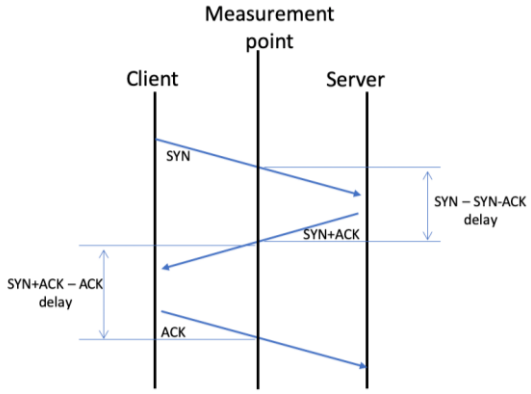

Figure 15 - Using the TCP Handshake to Measure Latency

6.2 Industry Measurement Initiatives

This section describes some of the commonly used measurement architectures.

6.2.1 SamKnows Whitebox (Dedicated Test Device Solution)

SamKnows has developed a "Whitebox," a dedicated device with a test suite, for measuring Internet performance. These Whiteboxes are used by service providers, government regulators, etc., and the tests can also be incorporated into network devices like modems or routers. The SamKnows test methodology includes many aspects of measuring consumer broadband performance: providing consumer volunteers with Whiteboxes to run tests on consumer Internet connections, the mechanism for collecting and aggregating the data, and finally, the format for presenting the data.

The following describes the overall latency measurement methodology followed by SamKnows Whiteboxes. As described in SamKnows literature, upon startup, the Whitebox runs a brief latency measurement to all measurement servers hosted by an operator or hosted by SamKnows on its behalf. The server with the lowest round-trip latency is selected as the target for all subsequent measurements.

Below are some of the latency-specific tests that the SamKnows Whitebox, or routers with the test functionality, can run, as described in the SamKnows documentation.

- The latency and packet loss (UDP) test measures RTT of small UDP packets between the Whitebox and a target test server. Each packet consists of an 8-byte sequence number and an 8-byte time stamp. The test operates continuously in the background and randomly distributes the sending of the packets over a fixed interval, typically 2,000 samples per hour. It then records the number of packets sent, the average round-trip time of these packets, and the total number of packets lost. The test uses the 99th percentile when calculating the summarized minimum, maximum, and average results on each Whitebox.
- The contiguous packet loss/disconnections (UDP) test records instances when two or more consecutive packets are lost to the same test server. Alongside each event, the test records the time stamp, the number of packets lost, and the duration of the event. By executing the test against multiple diverse servers, an operator can begin to observe server outages and/or disconnections of the user's home connection.
- The latency, jitter, and packet loss (fixed-rate UDP test) test uses a fixed-rate stream of UDP traffic, a bidirectional 64-kbps stream (representative of the G.711 voice codec), running between the client and test

nodes. The standard configuration uses 500 packets upstream and 500 packets downstream. The server and the client record the loss rate and the jitter observed. Jitter is calculated using the PDV approach described in [RFC 5481]. The 99th percentile is recorded and used in all calculations when deriving the PDV.

- The latency and packet loss (ICMP) test measures the mean round-trip time of ICMP echo requests in microseconds from the Whitebox to a target test node.

6.2.2 The M-Lab NDT (User Initiated)

M-Lab is a consortium of research, industry, and public-interest partners and provides an ecosystem for the open, verifiable measurement of global network performance. All of the data collected by M-Lab's global measurement platform are made openly available, and all of the measurement tools hosted by M-Lab are open source.

M-Lab defines a test suite known as Network Diagnostic Tool (NDT), which is a single-stream performance measurement of a connection's capacity for bulk transport (as defined in IETF's [RFC 3148]). The NDT reports upload and download speeds and latency metrics. The NDT is run by users to test their Internet connections. As described in [M-Lab NDT], when the test is run, the client attempts to pick the nearest server from the geographically distributed network of servers provided by the M-Lab platform. The test suite uses a 10-second bulk transfer from the server to the client. The server is instrumented with the TCP kernel instrumentation and captures several variables of the TCP state machine every 5 ms of the test. The NDT uses the TCP RTT samples as the latency data points and reports the difference between the minimum and maximum RTTs observed during a test run.

6.2.3 Quality Attenuation

The Broadband Forum has developed a framework for relating network and application performance called Quality Attenuation (ΔQ). [TR-452.1] defines a reference architecture and specifies requirements for measuring and analyzing quality attenuation on paths and sub-paths of a broadband network. It includes an overview of the theory and principles of Quality Attenuation, example use cases, and the measurement approach. Quality Attenuation is an approach to systems performance analysis that has applicability to broadband networks. It gives more insight than simply using speed test results as a proxy for quality of experience and application outcomes, and greater measurement fidelity of packet layer performance than simple min/average/max latency and jitter measurements.

ΔQ can decompose a round-trip time into separate constituent components, corresponding to various sources of performance degradation (packet loss/delay). These components are structural (architecture/design), network dimensioning (link speeds, etc.), and network load/scheduling related. The component elements of ΔQ are composable, i.e., they are both additive within an individual link to give its resultant performance and can be accumulated along the end-to-end digital delivery chain (e.g., between user device or CPE and application server in the cloud data center).

Quality Attenuation uses random inter-sample timing. This shows the whole distribution of delay (by the Arrival Theorem). It avoids issues of relative phase of samples, e.g., with round-robin scheduling might miss the jitter this adds. It also exposes short-lived variations whenever they occur (even when duration is less than sample interval) and immediately (if duration is greater than sample interval).

The S-component (serialization delay) of Quality Attenuation measurements is typically very small on modern, correctly functioning broadband links. However, it can still be very insightful. Measuring S reveals important details such as the difference between 4 bonded 1G bearers and a 4G slice of 10G, fragmentation/quantization effects, frame fragmentation, impact of time-slotted bearer allocation, and mismatched bonded or load-shared paths.

The Broadband Forum also describes methods to perform ΔQ measurements using common active single-sided two-way measurement protocols and the specific or optional features of those protocols, as well as how to configure those protocols (see [TR-452.2]).

6.3 Commonly Used Tools

6.3.1 Using Iperf and Netperf for Latency Under Load (Working Latency)

Iperf and Netperf are two open-source network performance benchmark tools that can be used to measure speed and latency under load (LUL) (also known as working latency).

A speed and LUL measurement system is shown in Figure 16, as described in [Ozer et al.]. The idle latency portion of the measurement uses an HTTP CURL request/response, which uses TCP as its transport protocol. The latency-under-load portion of the measurement uses Netperf's request/response test, which uses UDP as its transport protocol. The throughput portion of the measurement uses the Iperf3 open source measurement tool, which uses TCP as its transport protocol. Both measurements are run concurrently. The TCP-based data transfer will attempt to maximize its throughput up to the available provisioned capacity, potentially filling up node buffers along the network path, while the UDP-based request/response will attempt to complete its transaction competing against the load from the throughput measurement. In this fashion, the test is simulating the user's in-home experience where one user may be conducting a bulk data transfer (e.g., large file photo downloads) while another user is trying to complete quick requests/responses (e.g., real-time gaming).

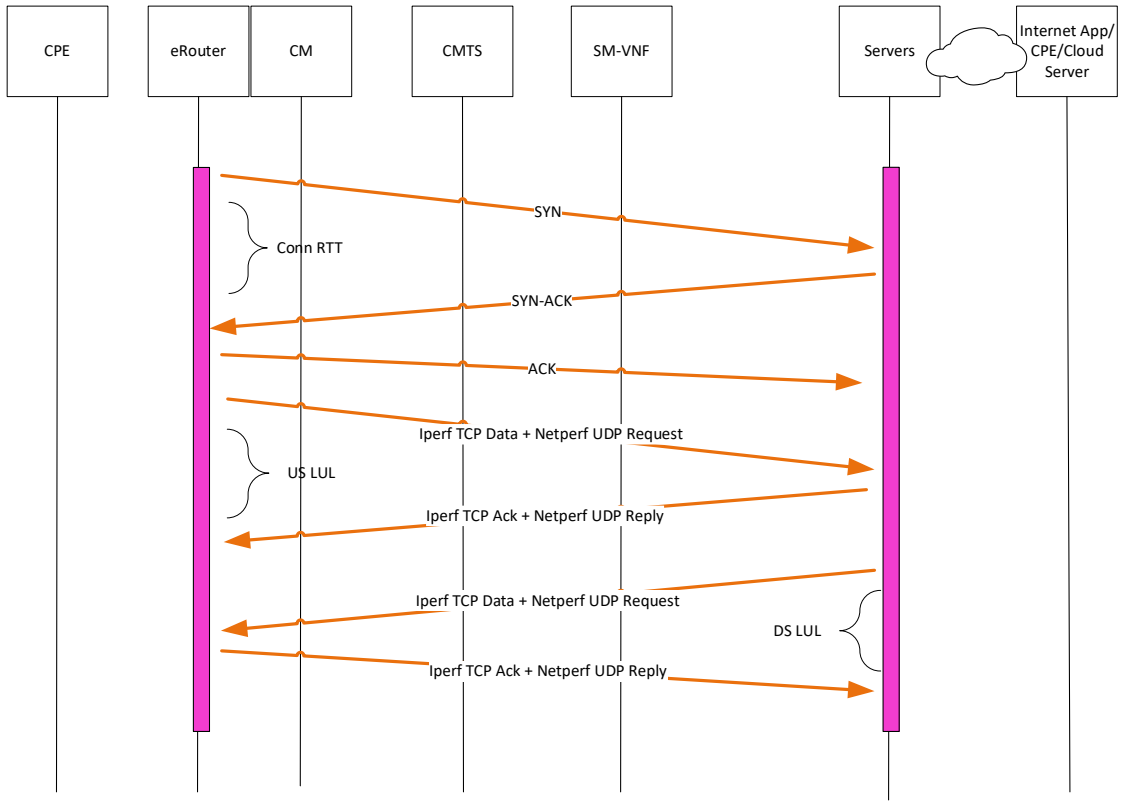

Figure 16 - Speed and Latency Under Load Measurements Using Iperf and Netperf

The latency reports can include min, mean, max, 50%, 75%, 95%, 99%, and standard deviation values. Packet loss can be estimated by monitoring the successful transactions. Methods such as IRTT can be extended for more accurate packet loss and latency values depending on an efficient implementation that can be supported by limited resources in the test devices or gateways. The accuracy and precision of measurements depend on the processing delays in the platform. The test parameters also affect the accuracy of the system latency. For example, to detect large queues on the path, the test parameters, such as test duration, number of TCP flows, and omit times, must be optimized to saturate the large queues because speed saturation may happen earlier than queue saturation.

Cable operators may run these measurements on test devices or integrate the firewall into their network devices. The accuracy of the system may be audited by third-party companies that independently review the test results generated by the platform [Jones et al.]. The measurement platform can be implemented with an embedded agent on cable modems based on the Reference Design Kit for broadband or OpenWRT. The measurement agent can interact with the control servers to process test requests using specific data plane test servers. The results can be reported back from the client and server. By launching the tests from within the modem itself, the network operator is able to measure as closely as possible to the in-home devices that utilize the Internet service. Different types of latency

values may be analyzed; e.g., an "idle" latency measurement, meaning no other concurrent traffic from the test user, and a "latency-under-load" measurement, meaning a latency measurement at the same time that a throughput measurement is conducted, where the throughput measurement is trying to maximize its data consumption. Comparing the idle latency vs. working latency measurements enables a clearer picture of the effectiveness of a deployed latency mitigation technique.

Operators may extend the system based on their needs by taking the following steps:

- measuring the impact of different TCP congestion control algorithms;
- implementing UDP-based data loading for speed test and working latency measurements, as QUIC protocols are widely used;
- marking test data to measure latency for different services, such as low-latency high-speed data flows; and
- exploring various control protocols to standardize test requests and results reporting.

6.4 Measurement Protocols

Two-way measurements are common in IP networks, primarily because synchronization between local and remote clocks is unnecessary for round-trip delay, and measurement support at the remote end may be limited to a simple echo function.

6.4.1 Two-Way Active Measurement Protocol (TWAMP)

[RFC 5357] specifies the Two-Way Active Measurement Protocol (TWAMP), which provides a common protocol for measuring two-way or round-trip measurement between network devices.

The [RFC 5357] TWAMP defines a standard for measuring round-trip network performance between any two devices that support the TWAMP protocols. TWAMP consists of two interrelated protocols: TWAMP-Control and TWAMP-Test. The TWAMP-Control Protocol is used to set up performance measurement sessions, i.e., to initiate, start, and stop test sessions. The TWAMP-Test Protocol is used to send and receive performance-measurement probes, i.e., to exchange test packets between two TWAMP entities. The TWAMP measurement architecture usually comprises two hosts with specific roles, as shown in Figure 17. The first host (controller) consists of the control-client, which sets up, starts, and stops TWAMP-Test sessions, and the session-sender, which instantiates TWAMP-Test packets that are sent to the session-reflector. At the second host (responder), the session-reflector reflects the measurement packet upon receiving the TWAMP-Test packet. The responder can also have the TWAMP server that manages one or more TWAMP sessions.

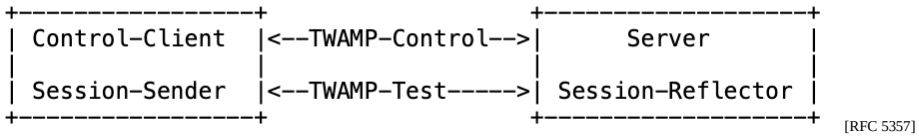

Figure 17 - TWAMP Reference Model

TWAMP Light is an alternative architecture that eliminates the need for the TWAMP-Control Protocol and assumes that the session-reflector is configured and communicates its configuration with the server through non-standard means. The session-reflector simply reflects the incoming packets back to the controller while copying the necessary information and generating sequence number and time stamp values. In TWAMP Light, the roles of control-client, server, and session-sender are implemented in one host (the controller), and the role of session-reflector is implemented in another host (the responder).

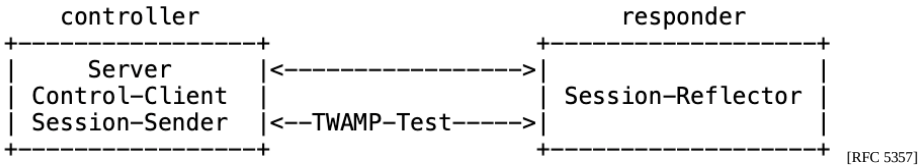

Figure 18 - TWAMP Light Reference Model

TWAMP is more accurate than simple ping or trace route measurements and is used by many operators in their transport, core, and access networks. Several independent implementations of both TWAMP and TWAMP Light [RFC 5357] have been developed and deployed and provide important operational performance measurements.

TWAMP is implemented in many of the core router products. TWAMP can provide accurate latency, jitter, and packet drop KPIs; is supported by many probe vendors; and can be integrated into network node equipment elements and CPE.

6.4.2 Simple Two-Way Active Measurement Protocol (STAMP)

Simple Two-Way Active Measurement Protocol (SWAMP) is a newer IETF standard [RFC 8762] that provides a simpler mechanism for active performance monitoring. It separates the control functions (vendor-specific configuration or orchestration) and test functions. STAMP also enables the measurement of both one-way and round-trip metrics (delay, delay variation, and packet loss).

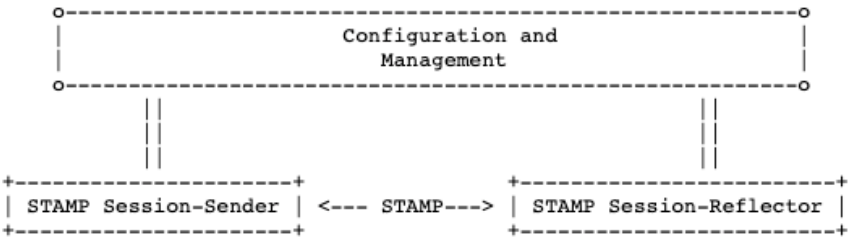

Figure 19 - STAMP Reference Model

STAMP is further detailed in Section 8.

7 MEASUREMENT CONSIDERATIONS

This section introduces a set of considerations that an operator must think through before building a latency measurement system.

7.1 Considerations for Conducting Latency Tests

7.1.1 Measurement Under Load vs. Quiet Times

Latency measurement tests need to ensure that testing is done over a variety of times to understand the variation between when the network is relatively lightly loaded and peak load time. This can also be used to measure self-congestion vs. network congestion. For example, measuring latencies at peak time in the evening when most of the subscribers on the plant are online is likely to catch incidents when the network is congested because of high overall network load. Another way latency numbers can be affected is the load within a single user's home itself or even within the same client. If multiple devices in the home are using the network for various purposes, like consuming video, voice calls, and gaming, then a latency measurement likely will yield different numbers than running the test at quiet times. The path to various servers can change automatically to accommodate network/routing changes; therefore, measured latency may vary over time, and it may be appropriate to get a broad picture of latency that includes such situations.

7.1.2 Window over Which the Measurement Is Done

Every latency measurement test can have a different purpose—one could be to obtain a quick and immediate diagnostic, whereas another could be to gather long-term statistics. To diagnose issues in the network, an operator will need to consider the correct amount of time to run a test, how many latency samples will be collected in each run, and how often the test will be run. This could include sample rates of once per hour, once per minute, once per second, and as frequently as 50 times per second. The sampling rate and the number of measurements run will depend on the ultimate goal of the operator. If the goal is to reflect the worst gaming experience, then more measurements that mimic the game traffic flows will offer a better idea of the performance of the network.

To understand latency, the entire distribution of latency measurements must be considered. Even though it is important for operators to look at latency numbers at the 99.9th percentile or higher, many monitoring systems stop at the 90th or 95th percentile. The reason is simply because it requires larger amounts of data to be collected, stored, and analyzed. The data collected by most monitoring systems is usually summarized in small 5- or 10-second windows. Given that it is not possible to meaningfully average percentiles or derive five nines from a collection of small samples of percentiles, there is no way to confidently know what the 99.99th percentile for the minute or hour was. A related question is how many total samples are needed to get valid statistics. If an operator wants to measure the 99.9th percentile latency, then at least 1,000 latency measurements are required, and a lot more (at least 2,000–8,000) would be needed to receive an accurate statistical estimate.

7.1.3 Off-Net and On-Net Testing

Active measurement architectures (e.g., SamKnows) may use client devices that run bandwidth and latency tests to a specific measurement server. A majority of test servers used by SamKnows customers are off-net, i.e., hosted on the Internet outside the operator network. Reporting results to target servers off an ISP's own network represents a real-world experience for end users. However, an ISP is not in control of the paths that get to the server and would also like to understand and debug issues within its own network. Therefore, many ISPs install test servers inside their networks ("on-net") to allow them to segregate on-net and off-net performance.

With both on-net and off-net servers in use, operators can see the difference in performance internal to their networks vs. external to them. The results can be used to troubleshoot peering links or routing issues or to simply rule out any capacity problems within the operator's own network. Consequently, any active measurement deployment should have a mix of on-net and off-net servers.

7.1.4 Sequential vs. Contemporaneous Measurements

An additional consideration is how measurements are made when there are multiple reflectors along the measurement path. There are two fundamental options: sequential vs. contemporaneous.

As an example, as shown in Figure 33 and Section 10.2, a measurement agent within a customer's or technician's client device can measure latencies to each of the session-reflectors within the gateway in their customer premises and can also run latency tests with the session-reflector close to the interconnection point.

With the sequential approach, the measurements would happen one after the other. With the contemporaneous approach, the measurements would measure to both reflectors as part of the same "measurement experiment;" for example, by capturing the delay/loss of individual packets accrued at each reflector point in the end-to-end connection path. This latter approach helps to give more accurate demarcation of any quality degradation issues in the connection because even though network topology and link speed are usually reasonably static (apart from occasional re-routes, etc.), the latency component of the network load can vary greatly at the sub-second level.

Applications like rate-adaptive video streaming adapt their encoding to "back off" the load they place on the network over a "long" period of a couple of seconds. However, millisecond-level variations in instantaneous load can impact individual packet delay volatility. This could be because of a performance issue that is more likely to be missed if sequential measurements are used.

7.1.5 Marked Traffic vs. Unmarked Traffic

Differentiated services, or DiffServ (see [RFC 2474]), specifies a simple mechanism for classifying and managing network traffic and providing quality of service (QoS) on modern IP networks. DiffServ can, for example, be used to provide low latency to critical network traffic, such as voice or streaming media, while providing simple best-effort service to non-critical services, such as web traffic or file transfers.

The six most significant bits of the DiffServ field (previously "Type of Service" (TOS) field) in the IP header are called the DSCP, and the last two bits are the explicit congestion notification (ECN) field. Routers at the edge of the network classify packets and mark them with their DSCP value in a DiffServ network. Other network devices in the core that support DiffServ use the DSCP value in the IP header to select a per-hop behavior for the packet and provide the appropriate QoS treatment. Various applications and services (typically UDP based) can also mark the traffic they generate with specific DSCP values. For example, the popular video conferencing application, Zoom, uses default DSCP marking values of 56 for audio, 40 for video, and 40 for signaling.

In Low Latency DOCSIS technology, by default, the traffic within an aggregate service flow is segmented into the two constituent service flows by a set of packet classifiers that examine the DSCP field and the ECN field. Specifically, packets with a non-queue building DiffServ value, 45, per a current [NQB PHB] draft, will get mapped to the low-latency service flow, and the rest of the traffic will get mapped to the classic service flow.

In the context of Low Latency DOCSIS and other technologies that support dual queue mechanisms, the question is how latency measurement tests can be modified to report metrics on unmarked traffic as well as marked traffic. One solution is to run any test twice—once as currently designed without any packet marking and once with marked DSCP packets—and report results on both. As more games and other applications start marking their packets, public Internet measurement reports will also have to start reporting latencies on both types of traffic.

7.1.6 Latency Measurement Test Definitions

When designing latency measurement tests, an operator needs to define the test and the associated parameters, such as the test traffic flow (i.e., the packet size and rate used), whether to test under load or without load, and the periodicity of the measurements.

Many IP network switches and routers need the full packet to be clocked into the device before it can be forwarded to the next networking device in the path to the end destination. This delay is referred to as a serialization delay, and these delays are often tested using 64-byte packets. For example, a 64-byte packet will have serialization delays of 5.12 µs when clocked in using a 100-Mbps port. However, serialization delays are usually proportional to the size of the packet. If the size of the packet was 1280 bytes, the serialization delays would be 20 times bigger at 102.4 µs. Even though this does not include the processing delays through a device (router, switch, CMTS, CM), it gives a sense of the interaction between packet size and link speed (interface bandwidth) that each node in the network could add as an absolute minimum.

Small (such as 64-byte) UDP packets sent every few seconds from a test node is a good place to start for RTT measurements. Latency tests that mimic the gaming experience (e.g., 150 Kbps upstream, 600 Kbps downstream, ~200-byte packets) would be a good dataset to collect to understand the impact on gaming or other real-time audio conferencing services. Latency tests with bigger size packets (1,500 bytes) could also be used to expose any packet size limitations in the network.

When testing latency, it is also a good idea to understand the latency when the network is under load vs. when it is not. Latency testing with load is typically done by running both downstream and upstream speed tests or something equivalent at the same time as doing latency measurements. While the speed test is running, the latency-under-load test can send packets to a target server and measures the round-trip time and number of packets lost. The test packets should be sent equally spaced over the duration of the speed test.

7.1.6.1 Latency Measurement Sampling Approaches

When designing latency measurement test cycles, an operator can take various time sampling approaches. The temporal sampling of latency for a particular device or an aggregate of devices will yield potentially interesting patterns in the latency behavior and variation through the day or through the week.

An operator could choose to run discrete latency measurement tests throughout the day, where each test runs for a short amount of time (e.g., a test of a few thousand packets may take a few seconds to run, and this test is repeated multiple times through the day at a predetermined frequency, such as every hour or every 15 min).

An alternate approach would be where an operator chooses to run a continuous set of no-load tests at a very low sampling rate and then add a few intensive tests on top of this to get additional latency measurements. A pseudo continuous test packet rate will be helpful with understanding any changes to routes within the network.

For a latency-under-load test, it also makes sense for an operator to figure out if the given service group is loaded (or the test server is loaded) and in that case program the measurement agent to reschedule the test.

Randomization in the start time of the test can reduce the risk of aligning with some network phenomenon; for example, multiple simultaneous tests may cause some correlation at an aggregation point.

7.1.6.2 Latency Measurement Test Packet Size

Most network links in service in broadband networks today have a latency characteristic that is not very sensitive to packet size. Therefore, for most baselining and ongoing testing, it is sufficient to perform measurements using a single packet size (typically, a small size should be chosen in order to minimize bandwidth utilization). That said, there are situations in which measuring latency as a function of packet size can be very useful in uncovering a misconfiguration or a potential optimization opportunity.

For that reason, it is recommended that network operators run ongoing latency monitoring testing using a 64-byte packet size and occasional additional (ideally concurrent) testing using 1500-byte packets to observe latency as a function of packet size. If the result of this observation indicates a significant change in latency relative to packet size, then it is recommended that the operator consider running latency testing with a range of packet sizes to build a fuller picture of the latency vs. packet size characteristic (packet sizes could be uniformly distributed between 64 and 1500 bytes or, alternatively, 5 selected packet sizes as described in [TR-452.1]).

This characteristic could, for example, show a constant slope that would indicate a serialization delay. Another characteristic might show breakpoints where latency undergoes a step change at particular packet sizes, which may indicate situations in which a multiuser link has short transmit slots that cause larger packets to be split across multiple transmission opportunities.

7.1.6.3 Latency Measurement Test Packet Intervals

A latency test will send out a set number of packets at a given rate or inter-packet interval.

Given that many real-time applications are on an isochronous schedule (e.g., a data packet every 10 ms or every 30 ms), it makes logical sense that a latency measurement test also try to use similar packet intervals to mimic the latency behavior for those types of traffic. Many existing test equipment will send packets at a fixed (user-configurable) interval.

Sending packets at a random time within a chosen inter-packet interval can also reduce the risk of the test aligning with some network phenomenon.

The timing for sending the source packets should be such that phase correlations are avoided, for example, random with an inter-sample interval following a negative exponential distribution which has the PASTA property (Poisson Arrivals See Time Averages) (see [TR-452.1]). Random number generators (e.g., from Python) can be used to generate the uniform distribution of packet sizes (selecting packet sizes from a finite, pre-determined set of values). Such random number generators can also be used to generate the Inter Packet Gap (IGP) with the appropriate negative exponential distribution.

It is also good to understand, for a given latency measurement test, whether the test should be implemented with random start times vs. random interarrival times (Poisson distribution). In both cases, the idea is to de-correlate mass tests, run on multiple devices on the network, from one another.

7.1.7 Measurement Accuracy: Timestamps

Latency measurements as described in this report invariably require comparing two timestamps, for example, the time when a particular packet is transmitted and when the corresponding response is received. In the case of active measurement protocols, some of these timestamps are actually embedded into the payload of the measurement packets and, in other cases, timestamps are produced as metadata associated with a packet (either by the operating system or by the application). In either case, it is important to understand the relationship between the timestamp value and the actual time at which the corresponding event occurred because discrepancies here will lead to noisy measurements.

If the timestamp is generated in the user-space application, there is the potential for significant discrepancies between the instant in time in which the system clock was queried and when the packet was actually transmitted or received. Much of this discrepancy is the result of the interaction between the NIC and the OS/kernel. It is common for high-speed interface NICs to support offloading features that are intended to reduce the kernel CPU usage. These features can result in hundreds of microseconds to milliseconds of delay (and/or variation) between when the application sends/receives the packet and when the packet was actually sent/received by the NIC.

In Linux systems, these offloading features can be disabled by installing and using a tool called "ethtool".

- Install ethtool: `sudo apt-get install ethtool`
- Turn off receive offloading on interface eth0: `sudo ethtool -K eth0 gro off`
- Turn off transmit offloading on interface eth0: `sudo ethtool -K eth0 gso off`

Turning off these offloading features can improve the accuracy of the timestamps and thus the accuracy of the latency measurement.

A better alternative is to use an NIC that supports hardware timestamping. On a Linux machine, timestamping options for the interface eth0 can be queried via ethtool: `sudo ethtool -T eth0`.

The various parameters are described in https://www.kernel.org/doc/Documentation/networking/timestamping.txt.

Via these features, a machine can get accurate (nanosecond) send and receive timestamps for packets.

When using tcpdump to receive packets, the available timestamping options can be queried, `tcpdump -i eth0 -J`, and can be selected via the -j option.

Definitions for the tcpdump timestamping options can be found in the pcap-tstamp man page (available online).

Timestamping support in Linux has improved markedly in recent years, but we were unable to find an open-source implementation that exploits this. An operator will need to implement this themself or validate an off-the-shelf implementation.

For a discussion on "phase precision" and clock drift between measurement points, please refer to the section on timing requirements in [TR-452.1].

7.2 Latency Measurement Test Definitions

In addition to this technical report, the CableLabs Latency Measurement working group has developed two other publications with a goal of defining a set of latency measurement test definitions.

The first publication defines a set of latency measurement tests focused on testing latency under load, mainly in a lab environment; see [LM-LULT]. Characterizing and benchmarking the latency performance of network devices in the lab before deployment is a part of understanding and characterizing new network devices and network technologies. This publication documents a set of latency measurement tests that an MSO can integrate into their network planning, testing, deployment, and operations.

The second publication develops a set of latency measurement test registry entries, modeled along the lines of [RFC 8912]; see [LM-TRSE]. This publication defines standardized tests that an operator could implement in production networks to monitor latency on an ongoing basis. These sets of tests are intended to be run in a production network to create a live monitoring of latency across the operator network. These test definitions and the data from these measurements will give operators a good view into the latency performance of the network.

7.3 Data Aggregation

This section will describe how an operator can aggregate latency data across all of the measurements in the network. It will also describe how the data is aggregated and how it is summarized to help an operator understand the latency of the network. Lastly, this section will describe how long it is useful to retain latency data.

An operator, after running a multitude of latency tests, will look to aggregate the latency metrics across "test" iterations. If each test calculates a percentile value (e.g., 99th percentile latency), these percentile values cannot be aggregated meaningfully across multiple test runs into percentiles. To calculate the percentile number (e.g., the 99th percentile latency) across all the tests (e.g., over a week), the operator will need to store the raw data (from each test) to do so. For example, one cannot calculate overall P99 from P99 values of 100 tests, though one can calculate the summary statistics such as max/min, etc. For the percentiles from each individual test, an operator can get a percentile, a maximum value, and a range of values. Another consideration is a subsampling of data from each test instead of from all the samples.

When running tests that calculate the histogram bin counts of latency, these counts can be aggregated. An operator could use these histogram statistics to estimate percentiles in the future, as the loss of information is relatively less. An operator can use other lossless compression techniques to store data.

A likely testing approach that many operators may take is as follows. Run continuous (not active-load) tests (low-rate) through the day/week, etc., and then run additional periodic intensive tests as well. In this case, aggregating data can dilute the P99 numbers, as the latency may be "good" most of the time. Data aggregation here is not just for all the continuous tests; it could be based on time of day, day of week, etc., or daily busy hour vs. special events. When aggregating data across different time zones, attention needs to be paid to normalize the time zone when combining data.

Data aggregation that uses data from low-rate tests may not have enough tests to give information on shorter windows of time. The low-rate tests are useful for longer term patterns (days/weeks, etc.). That are also useful for before and after type testing, e.g., when an operator makes network changes.

As for the intensive tests, should they be aggregated. The general idea is that the individual test statistics would be stored and, depending on the thresholds, alarms would be generated, etc.

For network-level aggregation for cable operators, it makes sense to use a DOCSIS service group level at the maximum. Also, it is important to keep the topology data of the test endpoints so that an operator can understand the patterns across different parts of the network. In the case of DOCSIS 3.1 technology with Low Latency DOCSIS, an operator may also choose to aggregate data across different service flows (e.g., LLSF vs. Classic SF). Another slice of aggregation would be across different user service tiers.

Also, when different measurement agents are each generating a new 5 tuple for each test (different servers/tests that each probe/agent runs), there can be a lot of different 5 tuples. Managing the different 5 tuples and consistently comparing them should be top of mind for an operator developing access network and core network metrics.

Some of the methodologies and data views that drive the data aggregation would be as follows.

- Aggregation of data from a single endpoint
  - View of the latencies from a single test
  - View of the latencies from a set of tests over time (e.g., an hour or a day or a week)
  - View of the changes in baseline latency for this endpoint—The idea here is to compare percentiles (e.g., 10 percentile points per Section 5.6.2).
- Aggregation of data from a node
  - View of the latencies from multiple endpoints within the same node
  - Configuration of alarm thresholds (multiple levels) to monitor adherence to SLAs
  - Data aggregation at this level helps an MSO identify where to put the edge compute resources and drive other network architecture decisions.
- Aggregation of data from a CMTS
  - View of the latencies from multiple endpoints within the same CMTS, across multiple nodes
  - The use case here would be to identify the top 10 nodes or worst offenders, etc., in an effort to root out any misconfiguration or network issues across the MSO footprint.

7.4 Path Stretch

Outside of the access network, geographic distances between endpoints become important in determining the overall round-trip latency. A user communicating with an edge server in their metro area would be expected to experience lower latencies than they would if they were communicating with a server on another continent. Moreover, networks are not typically line-of-sight, so the topological distance that signals take from one endpoint to the other and back can be significantly greater than what geography would predict.

The Center for Applied Internet Data Analysis (CAIDA) publishes statistics that include round-trip time measurements from its San Diego, CA, USA, location to its collection of "Ark" probes that are scattered around the world [CAIDA]. The measurements are binned into 100-km bins, and within each bin, the median and quartile RTT values are calculated. From these data, it can be gathered that the "typical" (i.e., median) RTT as a function of geographic distance is approximately 1.7× what would be predicted based simply on the distance and the propagation delay of fiber. This "path stretch" factor could be a useful metric to compare the network latency of an operator's core network infrastructure to typical values in other networks. If measurements of an operator's core network paths result in a path stretch significantly greater than 1.7, it may be an indication that suboptimal route choices have been made.

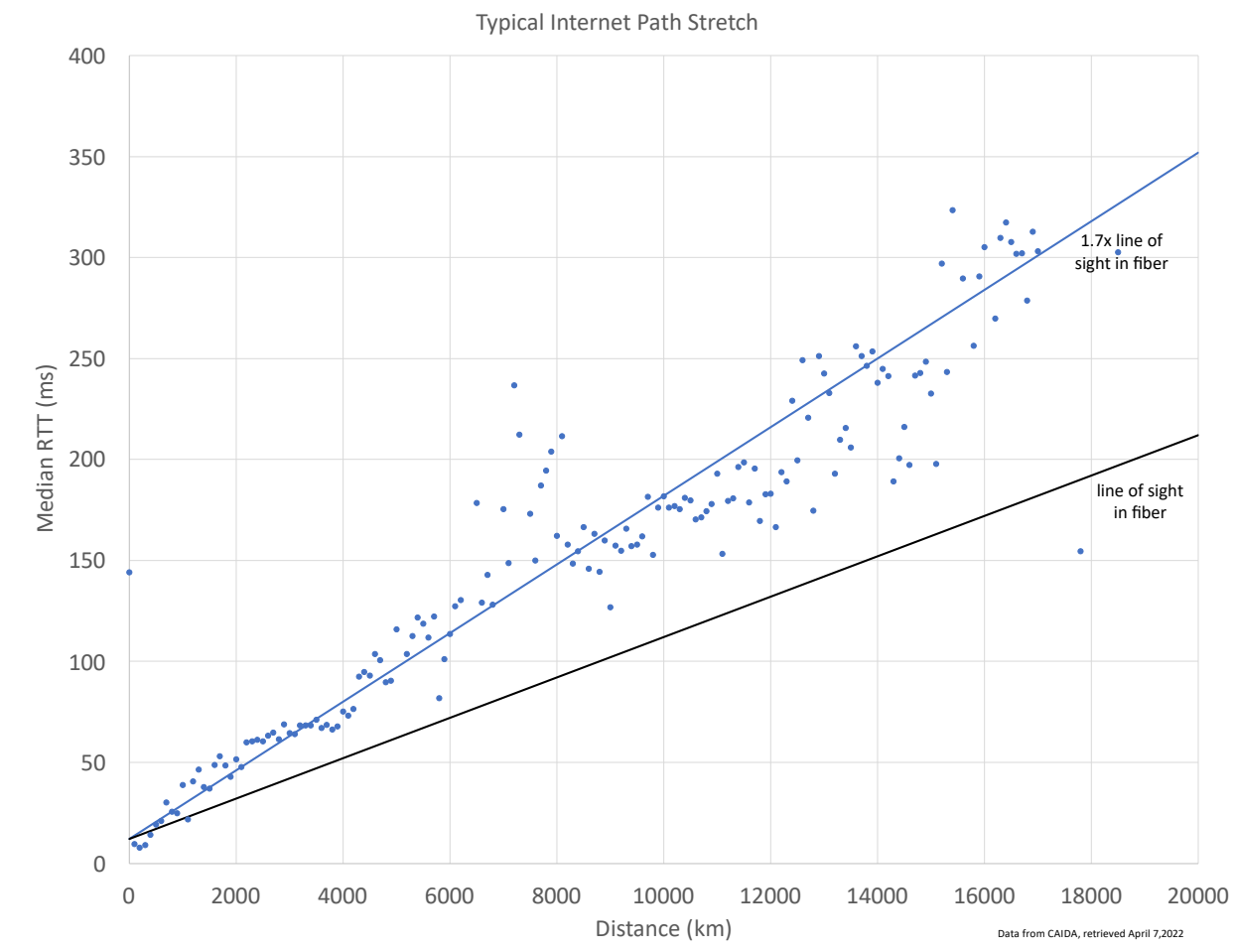

Figure 20 - Median RTT vs. Geographical Distance

Of note is that this typical RTT vs. distance value equates to approximately 60 km per millisecond.

8 SIMPLE TWO-WAY ACTIVE MEASUREMENT PROTOCOL

Simple Two-Way Active Measurement Protocol (STAMP)is an IETF-defined Standards Track [RFC 8792] that enables the measurement of both one-way and round-trip performance metrics like delay, delay variation, and packet loss.

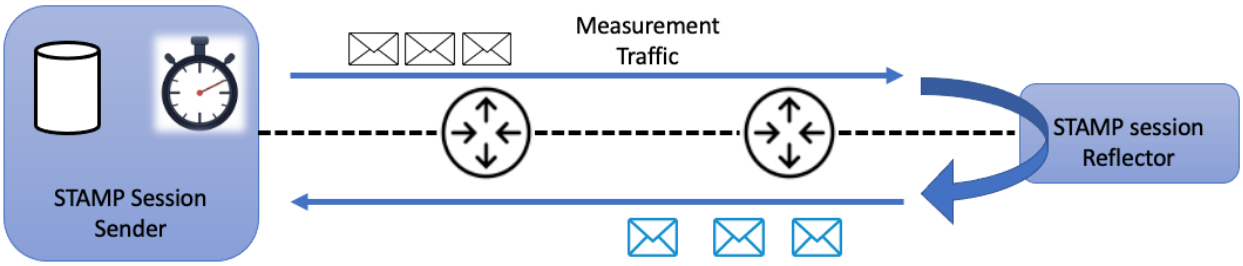

Figure 21 - Simple Two-Way Active Measurement Protocol

A STAMP measurement session is a bidirectional packet flow between a session-sender and a particular session-reflector for a given time duration. A STAMP session-sender transmits test packets over UDP to the STAMP session-reflector. The STAMP session-reflector receives the session-sender's packet and sends a response per the configuration.

The configuration and management of the STAMP session-sender, session-reflector, and sessions are outside the scope of [RFC 8792] and can be achieved through various means; for example, operators could develop command line interface, operational support system (OSS)/business support system (BSS), Simple Network Management Protocol (SNMP), and NETCONF/YANG-based software-defined networking (SDN) controllers.

8.1 Modes of Operation

There are two modes of operation for the STAMP session-reflector—stateless and stateful—per [RFC 8762].

For the stateless mode, the STAMP session-reflector does not maintain test state. It uses the value in the Sequence Number field of the received packet as the value for the Sequence Number field in the reflected packet. As a result, values in the Sequence Number and Session-Sender Sequence Number fields are the same, and only round-trip packet loss can be calculated while the reflector is operating in stateless mode.

For the stateful mode, the STAMP session-reflector maintains the test state. This enables the session-sender to determine in which direction the loss is happening by using the gaps between the Session-Sender Sequence Number and Sequence Number fields. As a result, both forward path loss and return path packet loss can be computed.

STAMP supports two authentication modes: unauthenticated and authenticated. Unauthenticated STAMP-Test packets ensure interworking between STAMP and TWAMP Light. As STAMP and TWAMP use different HMAC algorithms in authenticated mode, interoperability is only in the unauthenticated mode.

8.2 Port Number and Interop with TWAMP

A STAMP session-sender and STAMP session-reflector use UDP port 862 (same as TWAMP) as the default destination UDP port number. An implementation of the session-sender and session-reflector can define the port number to receive STAMP test packets from user ports and dynamic ports ranges.

One of the essential requirements of STAMP is the ability to interwork with a TWAMP Light device. For example, a TWAMP Light session-reflector may not support the use of UDP port 862, as specified in [RFC 8545]. A STAMP session-sender is allowed to use alternative ports. If any STAMP extensions are used, the TWAMP Light session-reflector will view them as the Packet Padding field.

8.3 Packet Format and Size

By default, STAMP uses symmetrical packets, i.e., the size of the packet transmitted by the session-reflector equals the size of the packet received by the session-reflector. The STAMP session-sender packet has a minimum size of 44 octets in unauthenticated mode (see Figure 22) and 112 octets in authenticated mode (see [RFC 8972]).

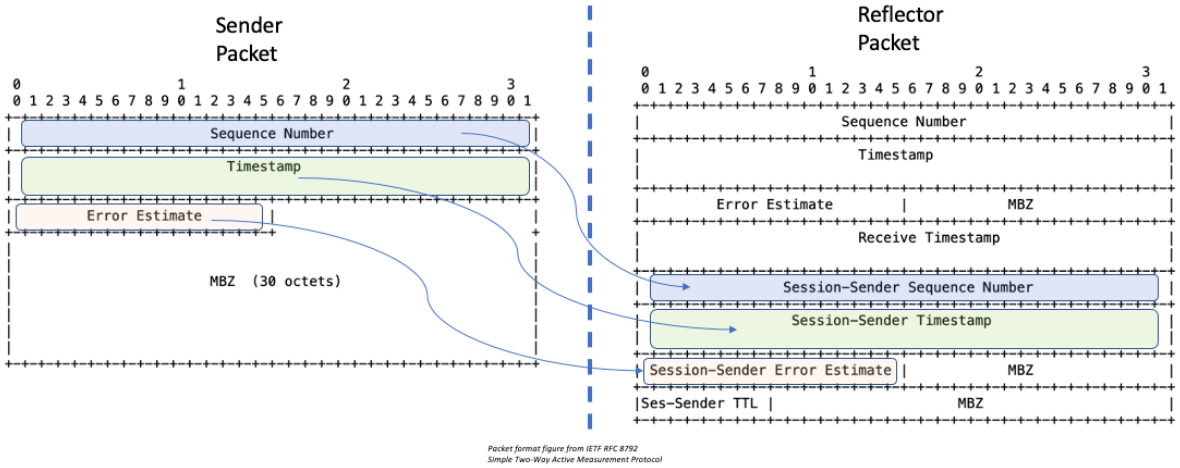

Figure 22 - STAMP Test Packet Format (Sender and Reflector)

STAMP supports a symmetrical size of test packets, i.e., a reflected-base test packet includes information from the session-reflector and, thus, is larger. To maintain the symmetry between base STAMP packets, the base STAMP session-sender packet includes the Must-Be-Zero (MBZ) field to match to the size of a base-reflected STAMP test packet.

Generating variable length of a test packet in STAMP is defined in [RFC 8972].

The field definitions for authenticated mode are the same as unauthenticated mode. The STAMP session-reflector test packet format in authenticated mode includes a hash-based message authentication code (HMAC) hash at the end of the protocol distribution unit (PDU). The detailed use of the HMAC field is described in [RFC 8762].

8.4 STAMP Extensions

STAMP defines multiple extensions to give the operator additional functionality regarding latency measurement. Some of these extensions include functionality such as extra padding, location, time stamp information, class of service, direct measurement, access report, follow-up telemetry, and HMAC. These functions are described in [RFC 8972].

Figure 23 shows the format of the type-length-values (TLVs) used within the existing STAMP packet [RFC 8972].

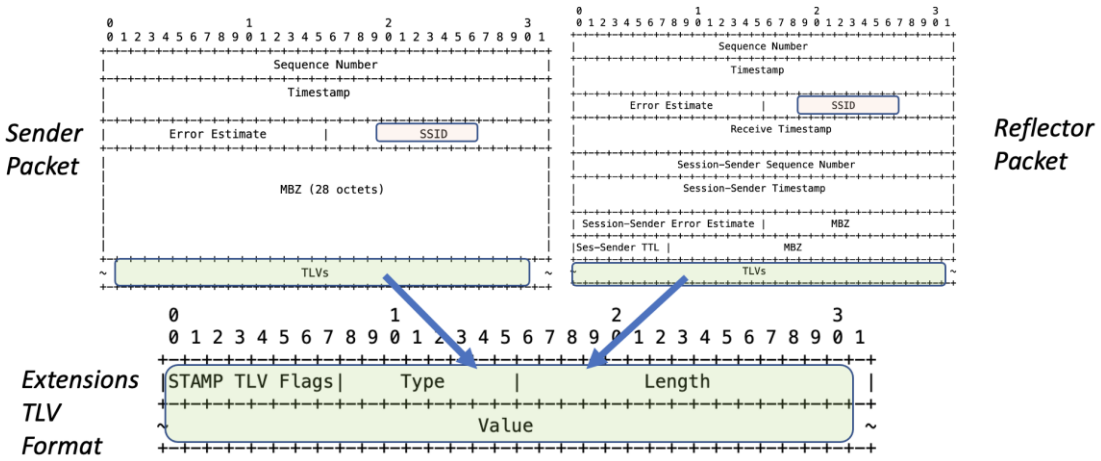

Figure 23 - STAMP Test Packet Extensions Format

IANA has created the "STAMP Sub-TLV Types" subregistry. The code points in this registry are allocated according to the IETF registration procedures [RFC 8126]: The range of 1–175 is under IETF Review, while the range 176–239 is designated as "First Come First Served", the range of 240–251 is designated for "Experimental Use", and the range of 252–254 is designated for "Private Use". Values 1–8 are defined for functions as shown in the figure below.

0	Reserved
1	Extra Padding
2	Location
3	Timestamp Information
4	Class of Service
5	Direct Measurement
6	Access Report
7	Follow-Up Telemetry
8	HMAC
255	Reserved

Figure 24 - STAMP TLV Extensions per [RFC 8972]

Of the eight types of STAMP extensions, defined in Figure 25, a couple are extensions. The "Extra Padding" TLV (Type 1) can extend the size of the STAMP packet to allow an operator to test the network with different-sized packets. The "Class of Service" TLV (Type 4) can be used by an operator to check how the DSCP value of the IP test packet changes as it traverses different networks.

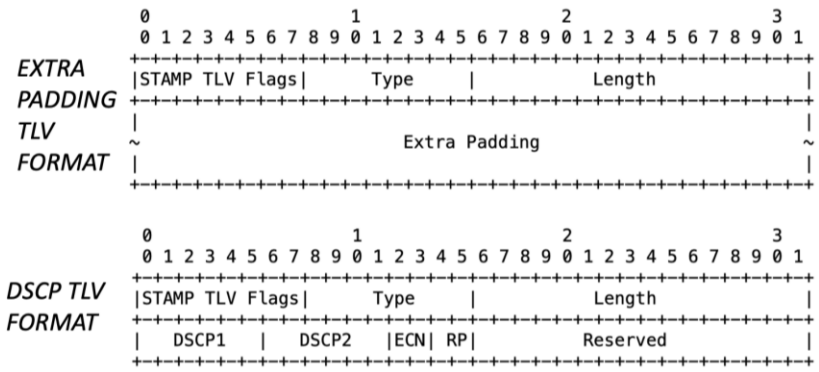

Figure 25 - STAMP Extra Padding and DSCP Extensions

8.5 New STAMP Extensions

The IETF has defined multiple extensions [RFC 8972] to the STAMP protocol to allow the use of optional informational elements within a STAMP packet for various purposes (see Section 8.4).

CableLabs has defined additional extensions that enable STAMP to traverse NATs, improve the scalability of STAMP session-sender functionality, better handle packet loss in the network, and be used for testing traversal of the ECN field. The definition of these extensions can be found in [LM-TRSE]. These extensions are meant to fit into and work with the existing STAMP technology.

8.6 STAMP Considerations

Below are some considerations when implementing STAMP; these issues need to be addressed by an operator looking to deploy STAMP.

In STAMP, for the authenticated mode, the minimum packet size is not that small (112 bytes in authenticated mode). This might constrain the usefulness of STAMP as a proxy for some highly interactive, small-packet-size applications (when used in authenticated mode).

In STAMP (and other similar protocols), the only record of the arrival of a test packet from the sender at the reflector is in the reflected packet. When the session-reflector is running in the "stateless mode" of STAMP, if a test packet is lost, then it is difficult for the sender to determine in which direction the loss occurred. When the session reflector is running in the "stateful mode" of STAMP, the direction of packet loss can be inferred by comparing the sent and reflected sequence numbers. The recommendation to operators is to implement the "stateful mode" of STAMP.

Another outcome of a lost reflected packet is that the associated outbound measurement (i.e., the latency timing from the sender to the reflector) is lost. Future STAMP extensions [RFC 8972] can be developed that could mitigate this issue.

A TWAMP Light client 'twampy' was found to misreport time and simply echo sequence numbers, rather than insert separate sequence numbers for the reflected packets, limiting the ability to test loss detection. This kind of problem also needs to be checked in STAMP clients.

STAMP RFCs do not specify the accuracy of the measurements. There is a need for conformance criteria (e.g., use of single timing point in reflection).

9 LARGE-SCALE MEASUREMENT OF BROADBAND PERFORMANCE

The Large-Scale Measurement of Broadband Performance [RFC 7594] developed by the IETF standardizes a measurement system for performance measurements of broadband access devices, such as home and enterprise edge routers, personal computers, mobile devices, and set-top boxes, whether wired or wireless.

Measuring portions of the Internet on a large scale is essential for accurate characterizations of performance over time and geography for network diagnostic investigations by providers and users. The goals are to have the measurements made using the same metrics and mechanisms for a large number of end points on the Internet and to have the results collected and stored in the same form.

9.1 LMAP Architecture

A large-scale measurement platform basically involves three types of protocols: a Control Protocol between a controller and the measurement agent (MA), a Report Protocol between the MAs and the collector(s), and several measurement protocols between the MAs and measurement peers (MPs) (used to perform the measurements). In addition, some information is required to be configured on the MA prior to any communication with a controller.

LMAP has defined the following protocols:

- a Control Protocol, for a controller to instruct measurement agents as to which performance metrics to measure, when to measure them, and how and when to report the measurement results to a collector; and
- a Report Protocol, for a measurement agent to report the results to a collector.

The LMAP framework has three basic elements: measurement agents, controllers, and collectors.

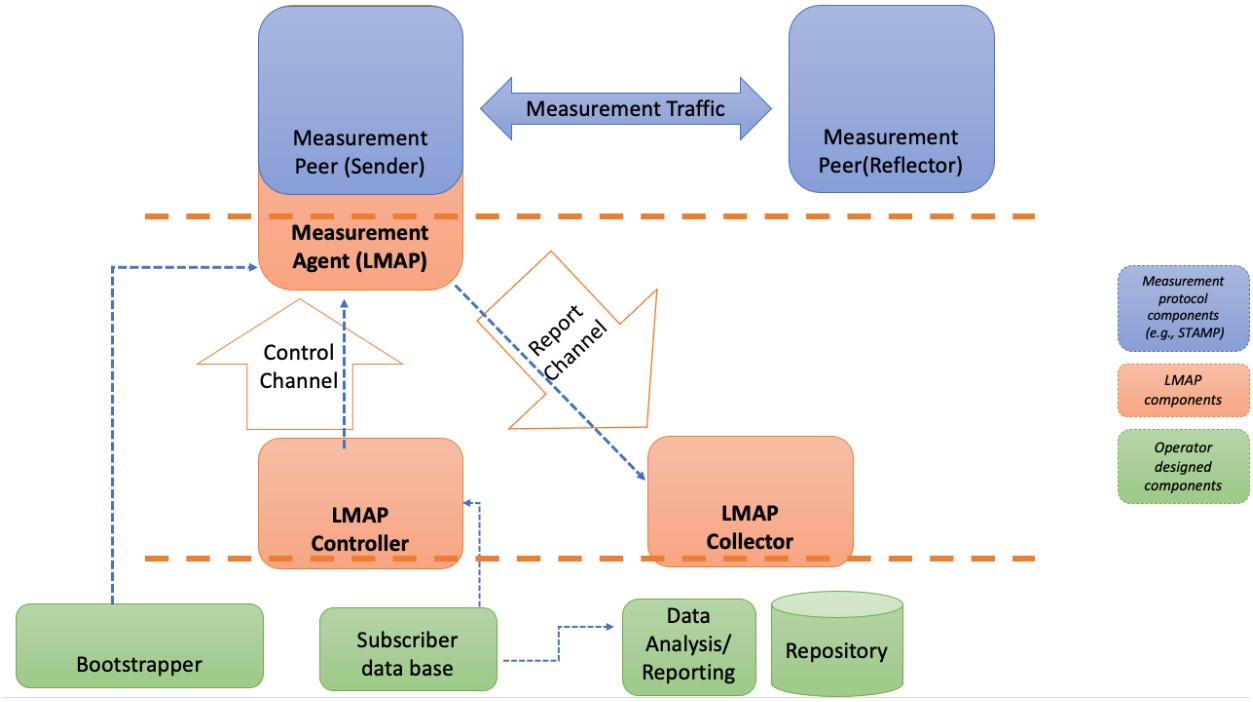

Figure 26 - Elements of an LMAP-based Measurement System

Measurement agents initiate the actual measurements, which are called measurement tasks in the LMAP terminology. In principle, there are no restrictions on the type of device in which the MA function resides.

The controller instructs one or more MAs and communicates the set of measurement tasks an MA should perform and when. For example, it may instruct an MA at a home gateway (for example, "Measure the 'UDP latency' with www.cableco.org; repeat every hour at xx.05"). The controller also manages an MA by instructing it on how to report the measurement results (for example, "Report results once a day in a batch at 4am") (a report schedule).

The collector accepts reports from the MAs with the results from their measurement tasks. Therefore, the MA is a device that receives instructions from the controller, initiates the measurement tasks, then reports to the collector. The communications between these three LMAP functions are structured according to a Control Protocol and a Report Protocol.

The LMAP effort has specified an information model [RFC 8193], the associated data models [RFC 8194], and protocols for secure communication. The information model applies to the measurement agent within an LMAP framework. It outlines the information that is configured on the measurement agent or exists in communications with a controller or collector within an LMAP framework. The purpose of such an information model is to provide a protocol- and device-independent view of the measurement agent that can be implemented via one or more Control and Report Protocols. The data models are extensible for new and additional measurements.

9.2 LMAP YANG Model

The LMAP framework has three basic elements: measurement agents, controllers, and collectors. Measurement agents initiate the actual measurements, called measurement tasks in the LMAP terminology. The controller instructs one or more MAs and communicates the set of measurement tasks an MA should perform and when. The collector accepts reports from the MAs with the results from their measurement tasks.

The YANG data model [RFC 8194] for LMAP has been split into three modules: The common module (ietf-lmap-common.yang) provides common definitions such as LMAP-specific data types. The control module (ietf-lmap-control.yang) defines the data structures exchanged between a controller and measurement agents. The report module (ietf-lmap-report.yang) defines the data structures exchanged between measurement agents and collectors.

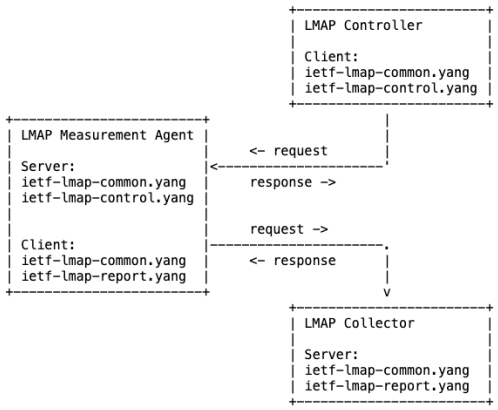

Figure 27 - High-Level View of the LMAP YANG Model Components

Figure 28 describes the various components within the LMAP-Control and LMAP-Report YANG data modules.

Figure 28 - High-Level View of the LMAP-Control YANG Model
YANG model view is built from tools at [LMAP YANG Tree].

The LMAP information model [RFC 8193] is divided into six functional parts mapped into the YANG data model as follows.

- Preconfiguration/bootstrapping information is outside the scope of the model.
- Configuration information is modeled in the /lmap/agent subtree, the /lmap/schedules subtree, and the /lmap/tasks subtree.
- Instruction information is modeled in the /lmap/suppressions subtree, the /lmap/schedules subtree, and the /lmap/tasks subtree.
- Logging information, i.e., success/failure/warning messages in response to information updates from the controller, will be handled by the protocol used to manipulate LMAP-specific configuration.

- Capability information is modeled in the /lmap/capability subtree. The list of supported tasks is modeled in the /lmap/capabilities/task list. Status information about schedules and actions is included in the /lmap/schedules subtree. Information about network interfaces can be obtained from the ietf-interfaces YANG data model [RFC 7223]. Information about the hardware and the firmware can be obtained from the ietf-system YANG data model [RFC 7317]. A device identifier can be obtained from the ietf-hardware YANG data model [YANG-HARDWARE].
- Reporting information is modeled by the report data model to be implemented by the collector. Measurement agents send results to the collector by invoking an RPC on the collector.

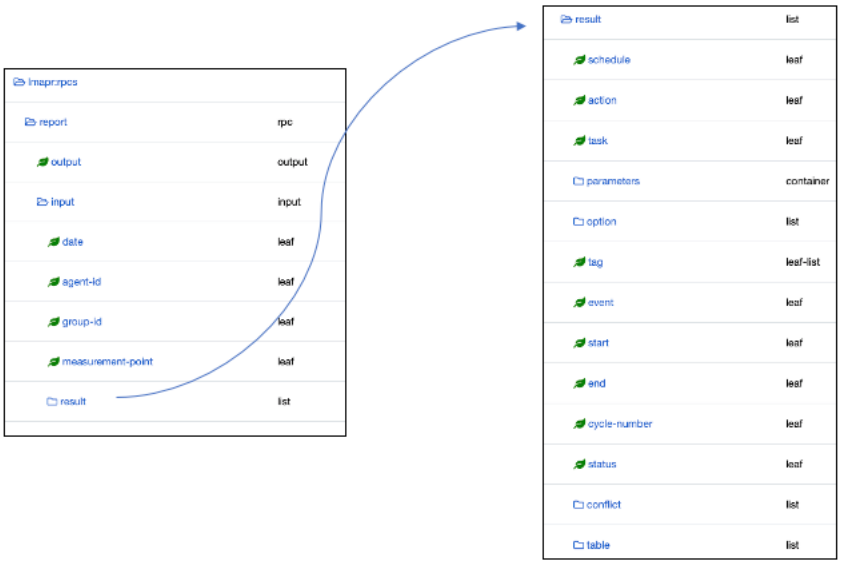

Figure 29 - Definition of the LMAP-REPORT YANG Model
YANG model view is built from tools at [LMAP YANG Tree].

10 LATENCY MEASUREMENT ARCHITECTURE

The latency measurement architecture can be split into two parts, as shown in Figure 30.

- Measurement domain includes the measurement protocol itself, the measurement agent, and the measurement peers. This domain performs the actual latency measurements/tests and calculates the latencies.
- Large-scale control and data collection includes the large-scale latency measurement orchestration across the network by a controller/collector entity. The controller entity coordinates the measurements across the various measurement agents in the network. The collector entity collects the data from the various measurement agents, then presents the aggregate data to the operator.

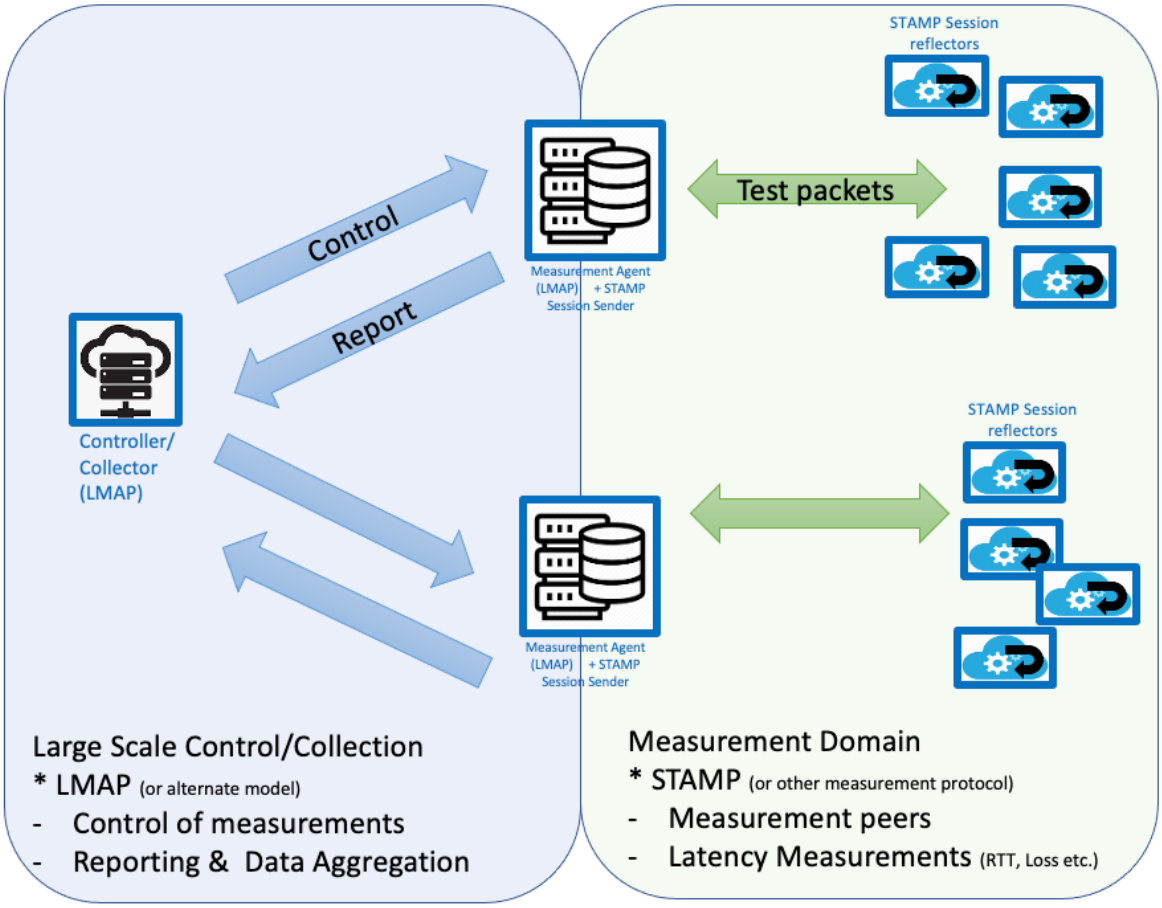

Figure 30 - Latency Measurement Architecture

Separating the data control and collection domain from the actual measurement domain allows the operator to make appropriate choices for both domains. The leading contender for large-scale control and collection is the IETF LMAP model. For the measurement domain, the IETF STAMP appears to be the best fit.

10.1 Measurement Architecture in a Cable Network

In a cable access network, the latency measurement architecture described above with an LMAP domain and the STAMP domain could be implemented as follows.

An operator could choose to deploy a measurement agent at each hub or headend location just north of the CMTS at that location. This way, an operator can reliably measure the latency on the DOCSIS/access network portion of the network. STAMP session-reflectors are lightweight entities and can be placed at the customer premises. This could be at the cable modem itself or on a gateway device that the operator installs at the customer premises.

To measure latencies in the core network, an operator could also place the lightweight STAMP session-reflectors at locations close to the interconnection/peering points to the Internet. Now an operator could deploy an LMAP collector and controller at a more centralized location, such as the network operations center, which oversees multiple hub/headend locations, or perhaps even the whole network of measurement agents (see Figure 31).

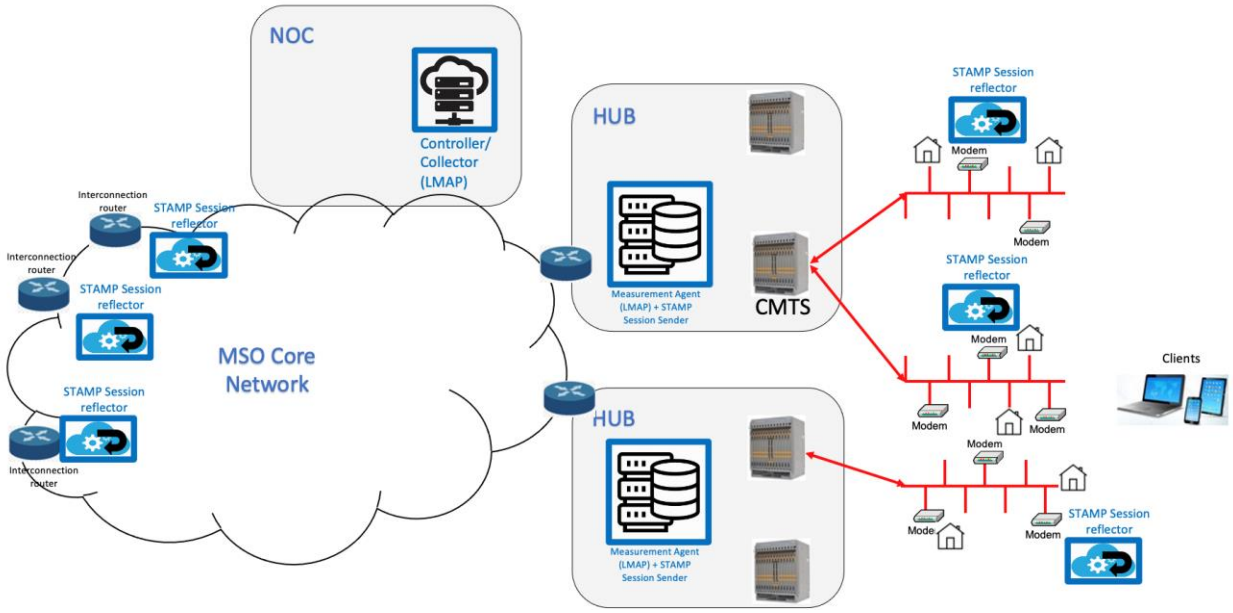

Figure 31 - Latency Measurement Architecture in a Cable Operator Network

10.2 Measurements in the Access vs. Core vs. Home Network

For the access network, each measurement agent at the hub locations is responsible for running latency measurement tests to each of the session-reflectors within its domain, i.e., within the part of the cable network to which it is connected.

For the core network, an operator could choose to run tests between every interconnection router and every measurement agent so that it can get a baseline understanding of the latencies across each of the potential paths across the core network.

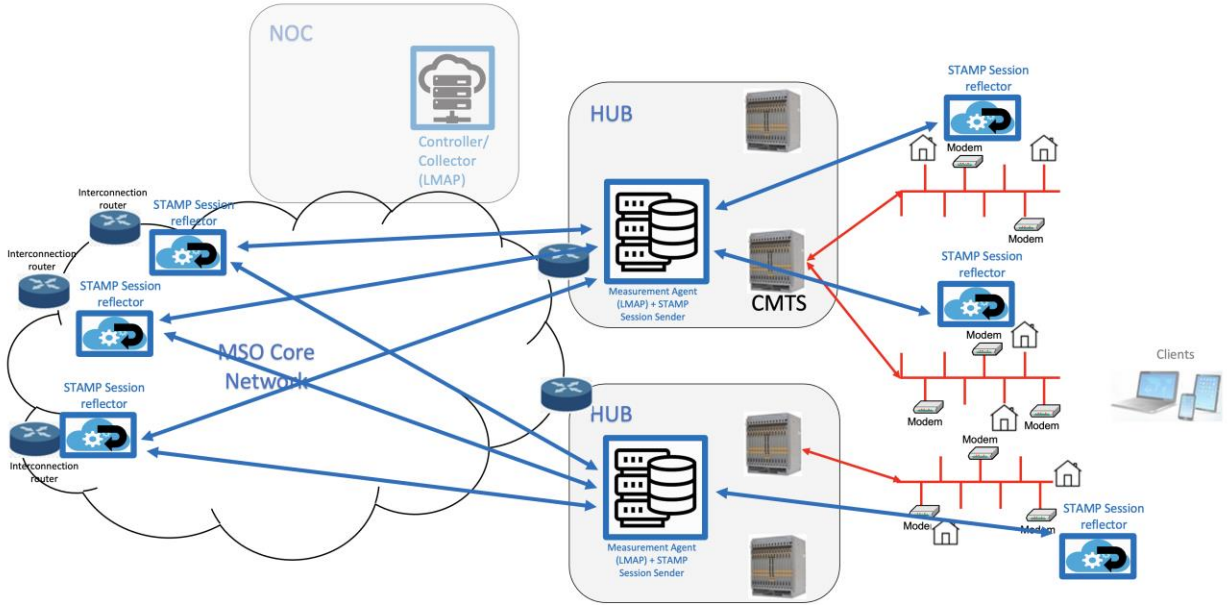

Figure 32 - STAMP Latency Measurements (Access and Core)

An operator can also instantiate a measurement agent within a client device, such as a handheld device or a laptop, which could be owned by a customer or by a technician. In this case, this measurement agent within the customer's home can help measure latencies to each of the session-reflectors within the network. In the case where this measurement agent talks to the session-reflector within the gateway in its customer premises, this will result in the operator understanding the latency in the home network (e.g., Wi-Fi). When the measurement agent in the handheld device runs latency tests with the session-reflectors close to the interconnection points, the operator can also get an understanding of the combined access and core network latencies from the customer location to the peering point.

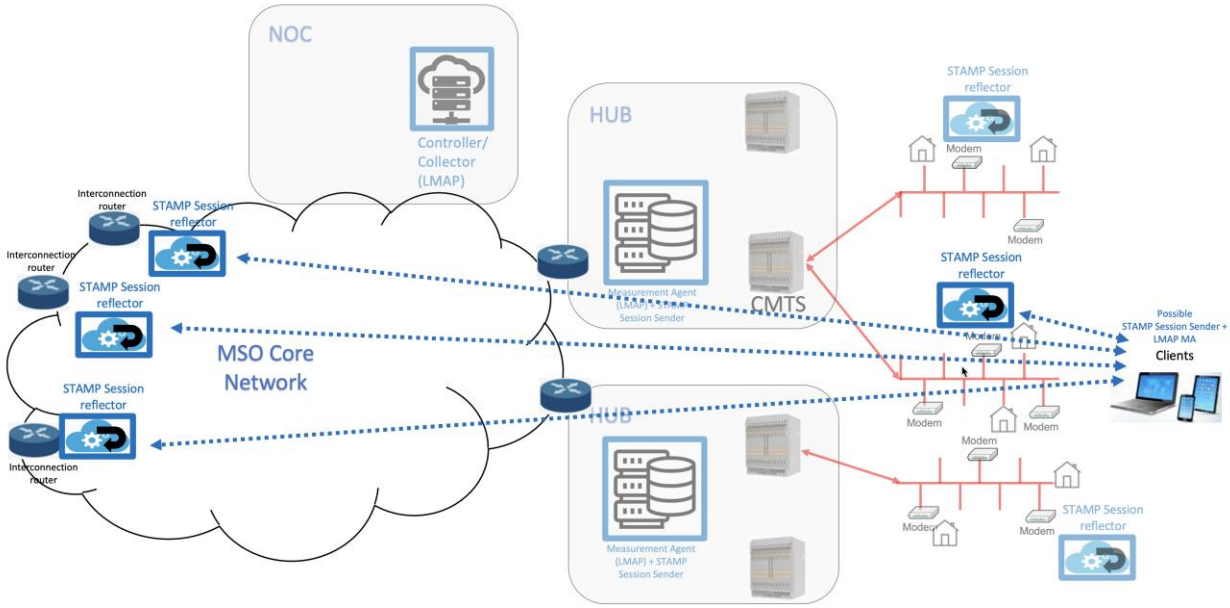

Figure 33 - STAMP Latency Measurements from a Client-side MA

10.3 Measurement Data Collection

For a large-scale measurement system, the LMAP controller/collector coordinates with each of the measurement agents in the network. The controller commands each of the measurement agents to run specific latency tests at a particular point in time or on a schedule. Each measurement agent collects the set of latency measurements and then computes the requested statistics, be it histogram counts or percentile data for that test. These results are reported back to the collector. This way, the LMAP controller/collector becomes the one central location where an operator can go to understand the latency performance across the whole network. All of the data analytics and data aggregation off the latency measurements across both the access network and the core network (and potentially the home network) will be implemented at the collector.

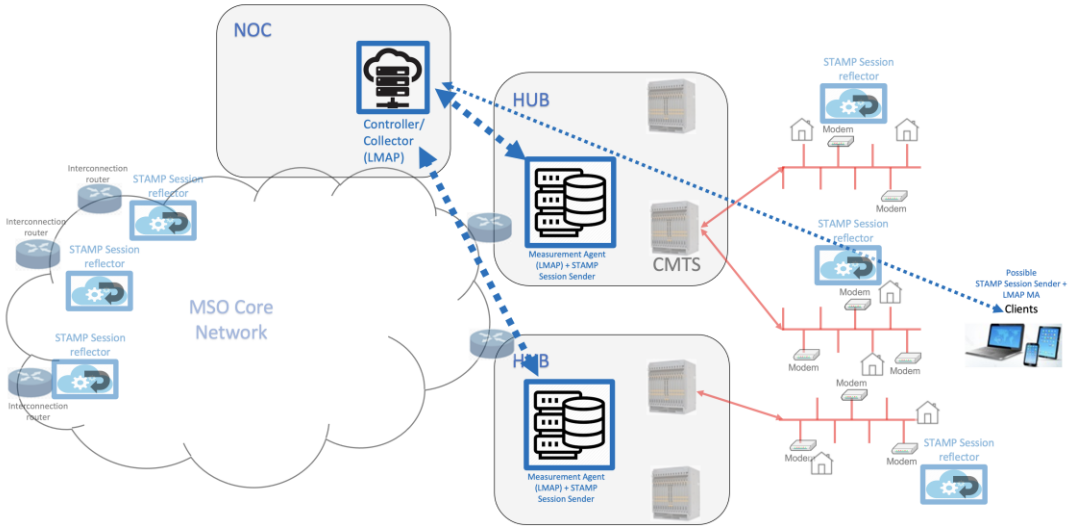

Figure 34 - LMAP Measurements Control and Reporting

10.4 Scaling Considerations

An operator would need to think through the scale of the measurement infrastructure that it deploys. The number of measurement agents and session-reflectors needed will depend on the size of the network. An operator would need to account for the number of headend/hub locations and the number of peering points it wants to cover.

10.4.1 Core Network Latency

As described above, for core network latency measurements, the operator would plan to have measurements from each hub or headend location to each of the Internet peering/transit points within MSO networks. In an informal survey, we found that each operator has somewhere between 8 and 12 peering points and transit locations, whereas the bigger operators have up to 30 interconnect locations (based on past mergers and acquisitions of cable properties). Typical operator networks vary from hundreds to thousands of CMTSs, and this implies tens to hundreds of CMTS hub/headend locations.

An operator would desire to get a full mesh of latency measurement of "CMTS" × "Interconnect" locations. This would be in the order of 100 links for a small cable operator and up to 1,000 links for a larger operator.

An operator will likely place one session-reflector at each interconnect location. An operator may choose to place one measurement agent at each hub/headend location. Alternatively, it could place it near the core/aggregation router to reduce the number of measurement agents, although this will lose the ability to truly isolate access network latency in the measurements.

10.4.2 Access Network Latency

Per the previous section, the assumption is that an operator places measurement agents at each hub/headend location. At this point, an operator needs to decide how many section-reflectors they want in the access portion of the network. An operator also needs to figure out if they want to perform latency measurements to every modem on the network or if they want to subsample the CM population.

If an operator decides to subsample the network, they need to decide what percentage of devices should be used in latency measurement, such as 20%, 10%, or 1%. For example, for a mid-tier operator with 10 million broadband subscribers, a 10% choice implies 1 million session-reflectors, and even a 1% sampling implies 100,000 session-reflectors. If an operator decides to equally distribute these session-reflectors across the CMTS footprint, assume that at 1,000 CMTSs with each CMTS supporting 50 nodes, this is just about two session-reflectors per node segment. These types of calculations give us an idea of the choices an operator will have to make in terms of the number of session-reflectors for the coverage of the modems that is needed and then start planning to scale the number of measurement peers accordingly. With distributed CMTS architectures (Remote PHY or Flexible MAC architecture technology) with Remote PHY and Remote MACPHY devices in the network, an operator may choose to measure those links separately, which means an additional layer of measurement agents at each of the nodes.

Once an operator gets comfortable collecting and understanding the latency measurements in a small part of the network, a phased approach to increasing measurement coverage across the network will be the likely path that operators take.

11 CUSTOMER EXPERIENCE

11.1 Literature Survey

This section is a literature survey on various ways to gauge customer experience based on the latency of their network connections. The following papers try to link the latency to the quality of experience of the user, especially for gaming applications. Table 3 summarizes the findings of some of these papers (indicated by an asterisk in the list below).

Gaming

- QoE and Latency Issues in Networked Games [link]*
  J. Saldana, M. Suznjevic, chapter in *Handbook of Digital Games and Entertainment Technologies*, Springer, 2015
- Cascading Impact of Lag on User Experience in Multiplayer Games [link]*
  E. Howard, C. Cooper, M. Wittie, S. Swinford, IEEE, 2014
- A Measurement Study Regarding Quality of Service and its Impact on Multiplayer Online Games [link]*
  M. Bredel, M. Fidler, IEEE, 2010
- How to Measure and Model QoE for Networked Games? A case study of World of Warcraft [link]
  M. Suznjevic, L. Skorin-Kapov, A. Cerekovic, M. Matijasevic, Springer, 2019
- OPScore, or Online Playability Score: A Metric for Playability of Online Games with Network Impairments [link]
  K. Gee, Ubicom, Inc., 2005
- Predicting the Perceived Quality of a First Person Shooter: the Quake IV G-model [link]
  A.F. Wattimena, R.E. Kooij, J.M. van Vugt, O.K. Ahmed, Association for Computing Machinery, 2006
- Empirical Study of Subjective Quality for Massive Multiplayer Games [link]
  M. Ries, P. Svoboda, M. Rupp, IEEE, 2008
- Networking and Online Games: Understanding and Engineering Multiplayer Internet Games [link]
  G. Armitage, M. Claypool, P. Branch, John Wiley & Sons, 2006
- Critical Sections in Networked Games [link]*
  S. Debroy1, M. Zubair Ahmad, M. Iyengar, M. Chatterjee, IEEE, 2013
- The Effects of Network Latency on Competitive First-person Shooter Game Players [link]*
  S. Liu, M. Claypool, A. Kuwahara, J. Scovell, J. Sherman, IEEE, 2021
- The Effects of Latency on Player Performance in Cloud-based Games [link]*
  M. Claypool, D. Finkel, IEEE, 2014
- Comparing the Effects of Network Latency vs. Local Latency on Competitive First Person Shooter Game Players [link]*
  S. Liu, M. Claypool, A. Kuwahara, J. Scovell, J. Sherman, EHPHCI: Esports and High Performance HCI, 2021
- Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2 [link]*
  R. Amin, F. Jackson, J.E. Gilbert, J. Martin, T. Shaw, Springer, 2013

Voice

- One-way transmission time, Recommendation G.114 (05/03) & Annex B ITU-T G.114 (05/2000) [link]
- Voice over IP Performance Monitoring [link]
  R.G. Cole, J.H. Rosenbluth, SIGCOMM Computer Communication Review, 2001
- DSL Forum Technical Report TR-126 Triple-play Services Quality of Experience (QoE) Requirements [link]
  T. Rahrer, R. Fiandra, S. Wright, Architecture & Transport Working Group, 2006
- Network Jitter: How It Affects Your VoIP Calls (+ How to Fix It Fast) [link],
  J. Manna, Nextiva blog, 2021
- Acceptable Jitter & Latency for VoIP: Everything You Need to Know [link]
  M. Grech, GetVoIP blog, 2018

Video Conferencing

- Zoom: Meeting and Phone Statistics [link]
- MS/Skype: Media Quality and Network Connectivity Performance in Microsoft Teams [link], Media Quality Summary Report in Skype for Business Server [link]
  Microsoft, 2021
- An Evaluation of Zoom and Microsoft Teams Video Conferencing Software with Network Packet Loss and Latency [link]
  T. Sel, A. Clopper, E. Baccei, Worcester Polytechnic Institute, 2020

Table 3 - Gaming Experience Summary from Selected Studies

Reference	Latency Takeaway	Year
QoE and Latency Issues in Networked Games	One-way delay of 80 ms could be acceptable for most of the users (cited 2005 paper). Quality level, rated by the players, dropped from “excellent” to “good” for one-way delays greater than 120 ms (cited 2008 paper).	2005–08
A Measurement Study Regarding Quality of Service and its Impact on Multiplayer Online Games	Armitage derives latency sensitivity estimates of users playing Quake III, finding that users do not connect to a server if round-trip times are above 150...180 ms. Henderson et al. state similar results of 225...250 ms for Half-Life. Delay and jitter have an especially negative effect on game experience, whereas loss has no measurable effect for values up to 40%.	2010
Critical Sections in Networked Games	We take the average RTTs for session lengths of 1 minute, which indicates the maximum duration of a typical critical section burst. We infer RTT values of above 150 ms as degrading and often notice spikes in various segments of critical sections.	2013
Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2	For the cases when the host is the uncongested user and the user under study is the congested user, a random jitter that is in a range of [0, 250] ms leads to a MOS less than 3. For the cases when the host is the congested user and the user under study is the congested user, a random jitter in the range of [0, 100] ms leads to a perceived MOS less than 3. The “expert,” or highly experienced, gamers significantly berate even the lowest level of network impairment (with constant 100 ms latency and no jitter).	2013
Cascading Impact of Lag on User Experience in Cooperative Multiplayer Games	The lag of just one player can cause a cascading impact on the QoE of other players. Our results illustrate the cascading impact of network lag through a statistical correlation between changes in QoE of the lagged player and of other group members.	2014
The Effects of Latency on Player Performance in Cloud-based Games	The results show cloud-based games are sensitive to even modest amounts of latency, with user performance degrading by up to 25% with each 100 ms of latency.	2014
The Effects of Network Latency on Competitive First-Person Shooter Game Players	Player performance and quality of experience both improve linearly as latencies decrease from 150 ms to 25 ms. Specifically, player accuracy at 25 ms is about 3% higher than player accuracy at 150 ms, and scores are 17% higher over the same range, an equivalent of about 1 additional kill or 2 additional assists per minute of gameplay. From 150 ms to 25 ms, QoE increases by about 25%, with the QoE at 150 ms being about 3.3 (on a 5-point scale) and the QoE at 25 ms being about a point better at 4.2.	2021
Comparing the Effects of Network Latency vs. Local Latency on Competitive First-Person Shooter Game Players	Based on observation of 68 users playing over 60 hours of CS:GO under controlled latency conditions, local latency has more impact on competitive FPS game players than does the same amount of network latency. In general, for a baseline system with a total of 125 ms of network latency and local system latency, a decrease in 100 ms of local latency improves accuracy by 6%, score by 3 points per minute, and QoE by 1.6 points on a 5-point scale, and a decrease in 100 ms of network latency improves accuracy by 2%, score by 2 points per minute, and QoE by 0.7 points on a 5-point scale.	2021

12 EXPERIMENTAL RESULTS

To put all the latency measurement theories and protocols to test, we are building an experimental latency measurement system prototype that is being tested with measurement server and peer locations across the Internet and also being tested with an operator in a production cable network. A future version of this report will include results from these proof-of-concept efforts.

12.1 Prototype Components

For the latency measurement system prototype, the following components were implemented: a measurement agent (STAMP session-sender) and a STAMP session-reflector. Additionally, an LMAP Controller + Collector, along with adding LMAP functionality to the measurement agent, are currently under development. These prototype components will be made available at [C3 CableLabs] after the development is complete.

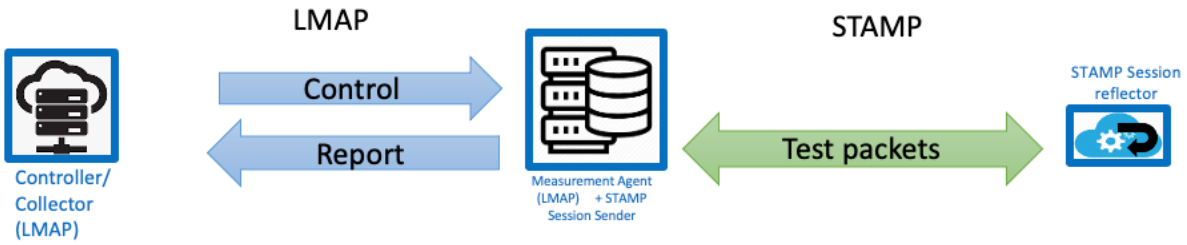

Figure 35 - Prototype Components

12.1.1 Session-Reflector

The STAMP session-reflector software implementation was developed using C and was built for a Unix environment. The Raspberry Pi platform was chosen for a session-reflector. The idea is to create a lightweight STAMP session-reflector software that could potentially be embedded within a cable modem or a gateway device in the house or a Wi-Fi access point. Another option would be standalone devices that the operator installs as "probes" throughout its network.

12.1.2 Measurement Agent

The measurement agent consists of two components: the STAMP session-sender and the LMAP measurement interface.

The STAMP session-sender software implementation was developed using C and was built for a Unix environment. The measurement agents were instantiated on Amazon Web Services (AWS) servers using Ubuntu Linux instances. The same version of the measurement agent was also deployed onto the Raspberry Pi platform to run locally to test and debug. The idea is to create a robust STAMP session-sender software that could potentially be extended to support all the STAMP features and support a variety of latency tests for an operator. This measurement agent (STAMP session-sender) will be controlled through the LMAP measurement interface it implements.

The LMAP measurement interface was developed to support a NETCONF interface and support the LMAP YANG model. Netopeer2 is an open-source server for implementing network configuration management based on the NETCONF Protocol. The server uses "sysrepo" (an open-source library for storing and managing YANG-based configurations for UNIX/Linux applications) as a NETCONF datastore implementation. The idea is that the measurement agent will interface with the LMAP controller using the NETCONF Protocol.

12.1.3 LMAP Controller and Collector

The LMAP controller and collector entity is being developed to support a NETCONF interface and support the LMAP YANG model. This communicates with the various measurement agents to configure tests and gather the results back. The additional data analytics and visualization happen here at the controller/collector.

12.2 Test Metrics

There are various metrics that an operator can track when it is looking to understand the latency performance of its networks, be it the access network, the home network, or the core network. Each portion of the network needs to be measured to understand the current characteristics and then to improve on it.

[LM SCTE 20] and [LM SCTE 21] discuss the basics of latency and propose some metrics to measure. The main measures that an operator should look at are latency RTT, PDV, and packet loss (and directionality of loss if available).

For the latency and PDV measures, if an operator were to pick a number, the 99th percentile value is a very good metric to track and is likely indicative of the customer experience, especially in latency-sensitive real-time applications. To understand how the latency behavior changes over time, it is also useful to track multiple percentile values; e.g., 0th percentile (minimum), 25th percentile, 50th percentile (median), 75th percentile, 95th percentile, 99th percentile, 99.9th percentile, and 100th percentile (maximum).

For the Delay-Loss RTT test, when reporting percentile latencies, the CableLabs Latency Measurement Test Registry Entries and STAMP Extensions Specification [LM-TRSE] defines a default set of percentiles consisting of 10 values corresponding to the 0th, 10th , 25th, 50th, 75th, 90th, 95th, 99th, 99.9th, and 100th percentile latencies. The test registry document also defines histogram bin edge definitions with 256 bins spanning from 0 ms to 3 s.

To understand how the latency varies over time and to visualize it, a simple time series graph shows a lot of interesting patterns. Additionally, a histogram of the dataset is a great place to start analyzing the latency performance. A cumulative distribution function (on a logarithmic scale) can show the operator the more interesting latency behavior regions.

13 CONCLUSION

Interactive applications, like gaming or real-time communication for which real-time responsiveness is required, are more sensitive to latency than bandwidth. These applications stand to benefit from technology that can deliver consistent low latency. Operators need to understand the latency characteristics of their networks and be able to delineate the latencies seen in the customers' homes, the access networks, and the MSO core networks. Using a common set of metrics to describe latency is the first step in understanding the state of the networks. Round-trip times are relatively easy to collect compared to one-way latencies. Multiple sets of measurements paint a better picture of the latency characteristics than single measurement. Using averages to measure latencies can be misleading, so an operator can choose better performance indicators, such as the 99th or 99.9th percentile, to track and understand latency behavior over time. Latency is currently being measured by national entities, raising awareness of the importance of operators to have their own latency measurement infrastructures. Active measurement techniques give an operator good control over the testing and a better understanding of the network over various times and conditions.

Latency measurement is a vital requirement for operators deploying new low-latency technologies going forward. Building a latency measurement system in hardware and software as a prototype can be relatively straightforward. Scaling it to production to measure the whole network requires planning and engineering efforts.

The most important first step is to implement a measurement protocol, and for this, we successfully used STAMP across the Internet for latency measurement. STAMP is lightweight and easy to implement on existing hardware and software platforms so that it can easily be deployed by operators either in a standalone Whitebox or as an add-on to existing devices (cable modems, gateways, or access points). STAMP offers a variety of functionality (e.g., round-trip and one-way measurements and loss, DSCP traversal, and different packet size testing) and can meet the needs of most latency measurement requirements.

The second main step is architecting the large-scale control and collection of data, and LMAP fits that bill quite well. The LMAP control and report architecture provide the operator with a well thought-out set of information/data models to initiate latency measurement and collect data at scale. Understanding the CDF of the latency measurement and tracking a set of percentile values should give an operator a very good understanding of latency performance of its networks and how they change as it deploys newer technologies.

Appendix I Acknowledgements

We wish to thank the following participants who contributed directly to this document.

Contributor	Company Affiliation
Karthik Sundaresan, Greg White, Steve Glennon, Jay Zhu	CableLabs
Gavin Young	Vodafone
Sebem Ozer	Comcast
Members of the Latency Measurement MSO Working Group	
Sebnem Ozer, Jesse VanLeemputte, James Martin, Allen Huotari	Comcast
Jud Whiteneck, Moutaz Elkaissi, Lei Zhou, Charles Cook, Shlomo Ovadia, Philip Anderson	Charter
Ed Heffernan, Nasir Ansari, Jeffrey Lee, Derek Lee, Richard Yee, Voltaire Aguda, Luis Silva, Amir Hanna	Rogers
Gavin Young, Bruno Cornaglia, Tino Muders, Kevin Smith	Vodafone
Amine Naak, Claire Perrier, Eric Menu	Videotron
Jeff Finkelstein, Michael Overcash, Dave Burns, Mark Adams	Cox
Joseph Nicksic	CableOne
Colin Dearborn, James Kerelchuk	Shaw
Robin Lavoie	Cogeco
Ameer Parab, Tomasz Kubinski, Deqing Mu, Jonathan Pannell,	Liberty Global
Par Mattsson	Tele 2
Klaus Dargel	PŸUR
James Andis, Ryan Wilkins	nbn
Karthik Sundaresan , Greg White, Steve Glennon, Stephen Froehlich, Barry Ferris, John Bahr, Volker Leisse, Matthew Schmitt, Jay Zhu	CableLabs

* * *