SCTE CABLE-TEC EXPO FAST FORWARD 2021
VIRTUAL EXPERIENCE
OCTOBER 11-14

2021 Fall Technical Forum
SCTE • NCTA • CABLELABS

A Latency Measurement System Using STAMP

with LMAP for large scale data collection

A Technical Paper prepared for SCTE by

Karthik Sundaresan
Distinguished Technologist
CableLabs
858 Coal Creek Circle, Louisville, CO
3036613895
k.sundaresan@cablelabs.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

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

This paper aims to share the experience from developing a simple end-to-end latency measurement framework. A new measurement protocol defined in the IETF is STAMP (Simple Two-Way Active Measurement Protocol, RFC 8762). The paper will provide the lessons learnt 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 small number of measurement entities and scale up as needed. If a Session reflector can by dynamically instantiated in a gateway, then one can run measurements on-demand. The paper 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 LMAP (Large-Scale Measurement of Broadband Performance). The paper 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. The paper also reports on a prototype STAMP measurement system which is deployed and collecting latency data.

1.1. Latency Metrics

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. Round trip time (RTT) or round-trip 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, plus the total length of time it takes for receiver to send a packet back to the sender, through one or multiple hops.

Packet Delay Variation, PDV, 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.

1.2. Latency Measurements

Active measurements are conducted by generating traffic between two end points for the sole purpose of measuring the latency. 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.

Active measurements are conducted by generating traffic between two end points for the sole purpose of measuring the latency. An Active measurement method depends on a dedicated measurement packet stream and observations of the packets in that stream. These packets are used to measure packet delay, and packet loss. One-way performance metrics need clock synchronization across the test points for measurements which are tough to implement. Active measurements can use protocols such as TWAMP/ TWAMP Light/STAMP.

1.3. Measurement Protocols

[IETF 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 inter-related protocols: TWAMP-Control and TWAMP-Test. TWAMP Light is an alternative architecture which 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.

Simple Two-way Active Measurement Protocol (STAMP) is a newer IETF standard [IETF RFC 8762] which provides a simpler mechanism for active performance monitoring. It separates the control functions (vendor-specific configuration or orchestration) and test functions. STAMP enables measurement of both one-way and round-trip metrics (delay, delay variation, and packet loss). It is intended to be used on production networks to enable the operator to assess service level agreements based on delay, delay variation, and loss.

2. STAMP

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

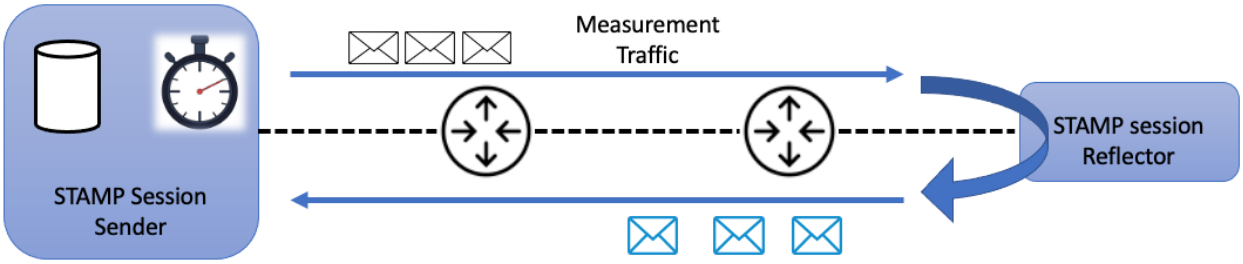

Figure 1 – Simple two way Active measurement protocol

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 the STAMP RFC and can be achieved through various means. E.g., Operators could develop: Command Line Interface, Operational Support System (OSS) / Business Support System (BSS), SNMP, and NETCONF/YANG-based Software-Defined Networking (SDN) controllers.

2.1. Modes of operation

There are two modes of operation for the STAMP Session-Reflector, stateless and stateful per [IETF RFC 8762]

Stateless: 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.

Stateful: The STAMP Session-Reflector maintains the test state. This enables the Session-Sender to determine which direction the loss is happening by using the of gaps 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.

2.2. Port number & 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 to 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 [RFC8545]. A STAMP Session-Sender is allowed to use alternative ports. If any of STAMP extensions are used, the TWAMP Light Session-Reflector will view them as the Packet Padding field.

2.3. 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 2) and 112 octets in the authenticated mode (see [IETF RFC 8972]

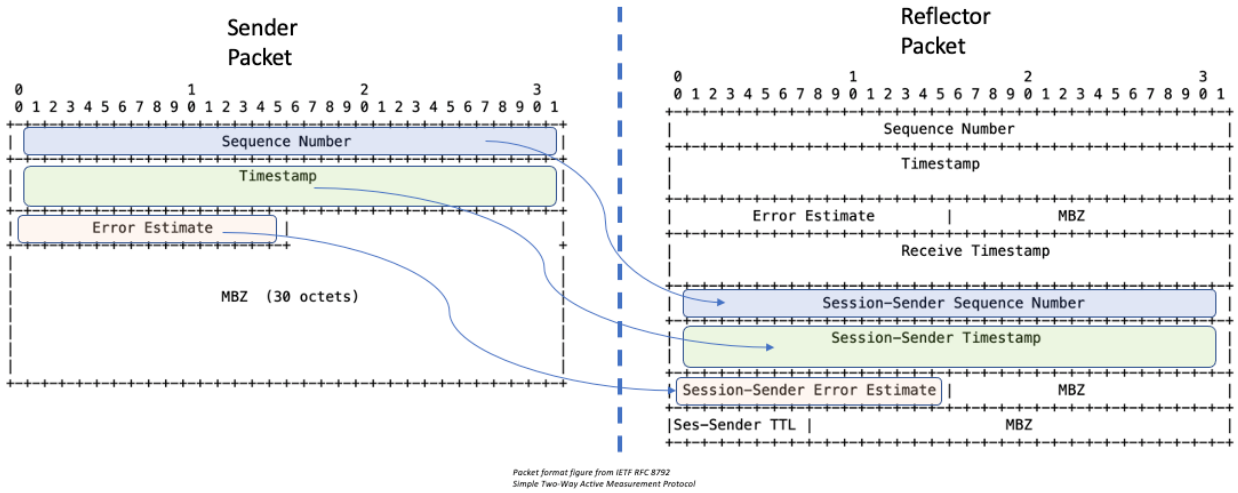

Figure 2 - STAMP Test Packet Format (Sender & Reflector)

STAMP supports 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 [IETF RFC 8972]

The field definitions for Authenticated mode are the same as the unauthenticated mode. The STAMP Session-Reflector test packet format in authenticated mode includes a HMAC hash at the end of the PDU. The detailed use of the HMAC field is in described in the [IETF RFC 8762].

2.4. STAMP Extensions

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

Here is the format of the TLVs used within the existing stamp packet. [IETF RFC 8972]

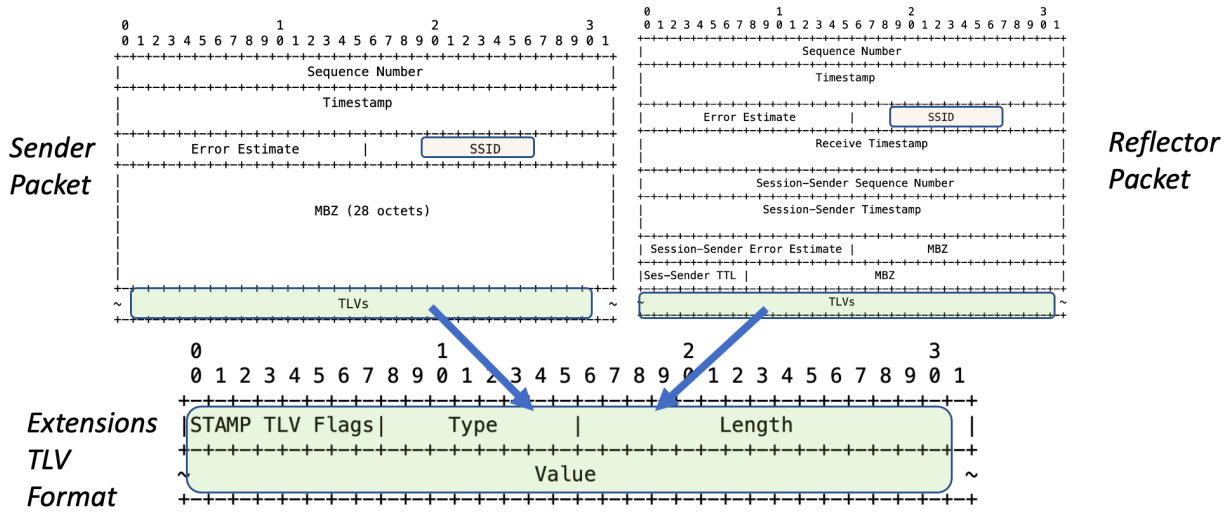

Figure 3 - STAMP Test Packet Extensions Format

Of the eight types of STAMP extensions, defined below are a couple of the 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.

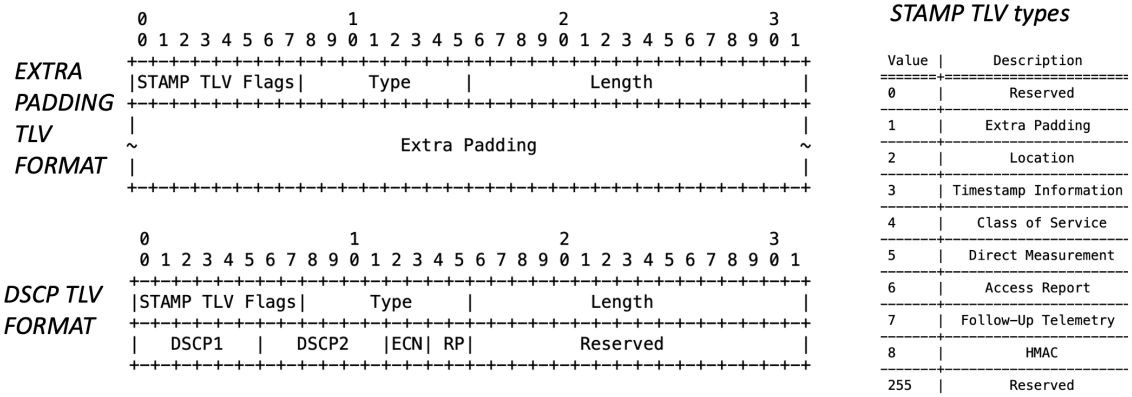

Figure 4 - STAMP Extra padding and DSCP Extensions

3. LMAP

The Large-Scale Measurement of Broadband Performance (LMAP) [IETF 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, set top box, 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 goal is 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.

3.1. LMAP Architecture

A large-scale measurement platform involves basically three types of protocols, namely, 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

- A Control Protocol, from a Controller to instruct Measurement Agents what performance metrics to measure, when to measure them, how/when to report the measurement results to a Collector.
- A Report Protocol, for a Measurement Agent to report the results to the Collector.

The LMAP framework has three basic elements: Measurement Agents, Controllers, and Collectors.

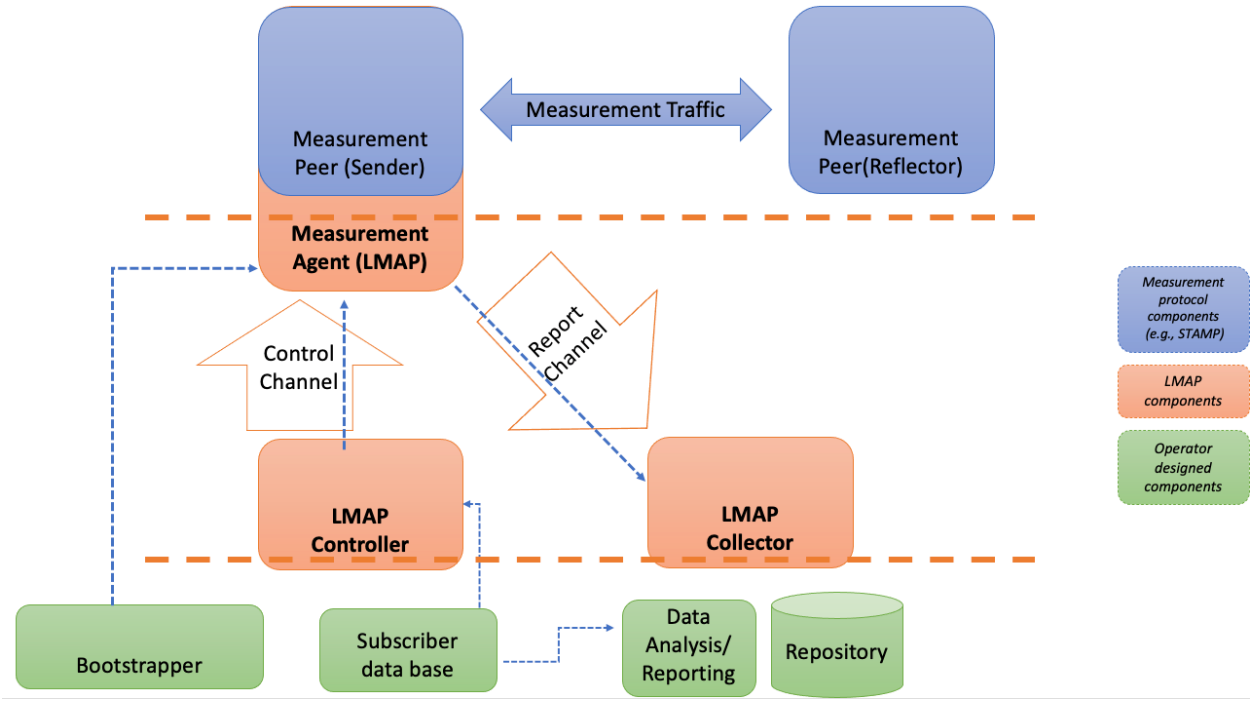

Figure 5 - Elements of an LMAP-based Measurement System

Measurement Agents (MAs) 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: "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 gets Instructions from the Controller, initiates the Measurement Tasks, and 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 [IETF RFC 8193], the associated data models [IETF RFC 8194], and protocols for secure communication. 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.

3.2. LMAP YANG model

The LMAP framework has three basic elements: Measurement Agents (MAs), 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 [IETF 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.

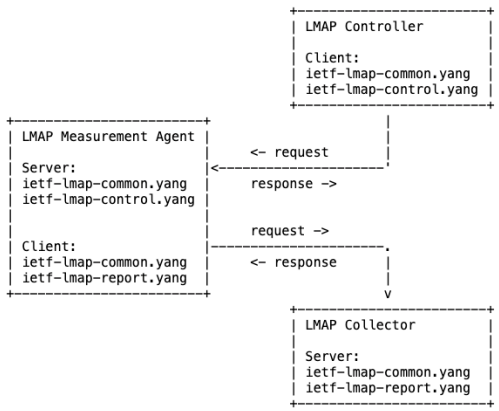

Figure 6 – High level view of the LMAP YANG model components

Below is a figure which describes the various components within the LMAP- Control and LMAP-Report YANG data modules.

Figure 7 – High level view of the LMAP-Control YANG model

(*) Figure 7 is built from views with the tools at https://yangcatalog.org/yang-search/yang_tree/ietf-lmap-control@2017-08-08

The LMAP Information Model [IETF RFC 8193] is divided into six functional parts mapped into the YANG data model as follows:

- Preconfiguration Information: bootstrapping information is outside the scope of the model
- Configuration Information: Modeled in the /lmap/agent subtree, the /lmap/schedules subtree, and the /lmap/tasks subtree

- Instruction Information: Modeled in the /lmap/suppressions subtree, the /lmap/schedules subtree, and the /lmap/tasks subtree.
- Logging Information: 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 and Status Information: Capability 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 [IETF RFC 7317]. A device identifier can be obtained from the ietf-hardware YANG data model [YANG-HARDWARE].
- Reporting Information: This 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.

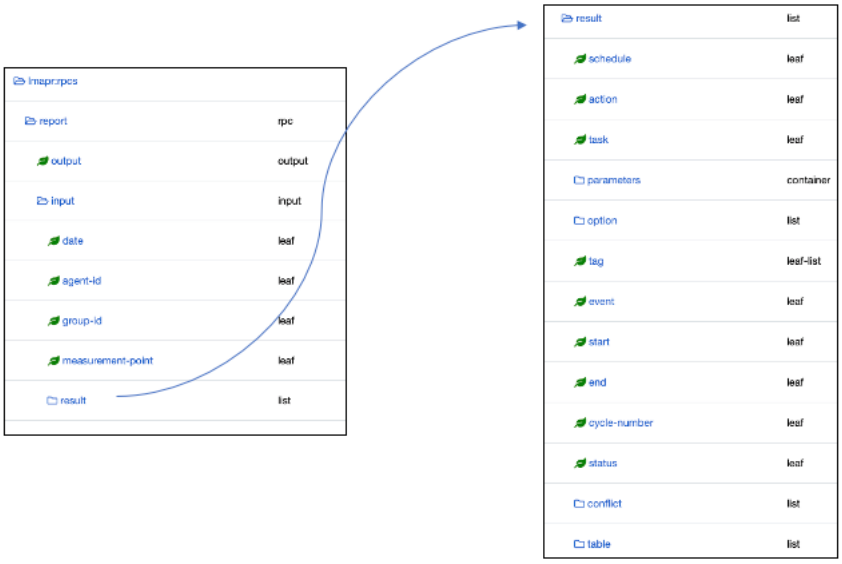

Figure 8 –Definition of the LMAP-REPORT YANG model

(*) Figure 8 is built from views with the tools at https://yangcatalog.org/yang-search/yang_tree/ietf-lmap-report@2017-08-08

4. Latency Measurement Architecture

The latency measurement architecture can be split into two parts as shown in the figure below:

Measurement Domain: This includes the measurement protocol itself and the measurement agent and the measurement peers. This domain performs the actual latency measurements/tests and calculate the latencies.

Large Scale control and data collection: This includes the large-scale latency measurement orchestration across the network by a controller/collector entity. The controller entity which coordinates the measurements across the various measurement agents in the network. The collector entity collects the data from the various measurement agents and then presents the aggregate data to the operator.

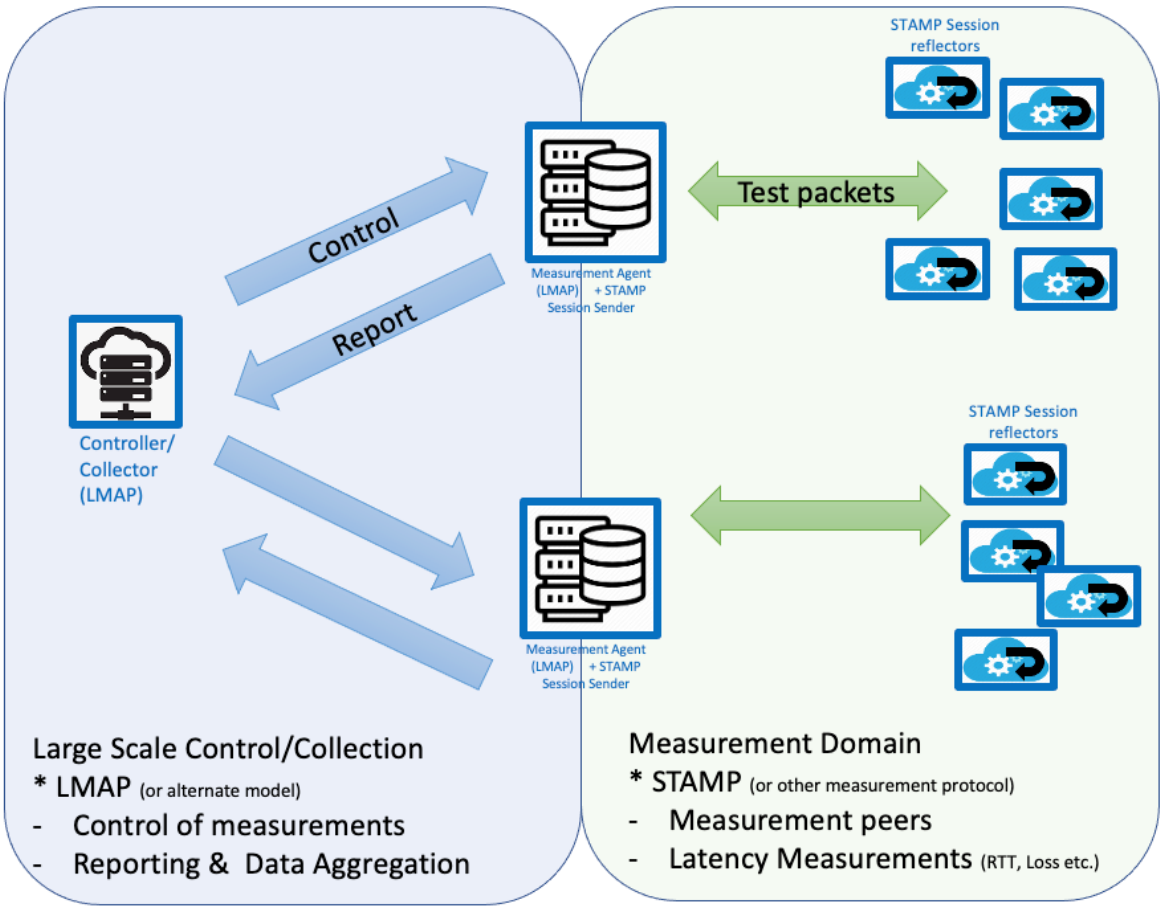

Figure 9 - Latency Measurement Architecture

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 measurement protocol appears to be the best fit.

4.1. Measurement Architecture in a Cable Network

In a cable access network, the latency measurement architecture described above with and 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 premise. This could be at the cable modem itself or on a gateway device which the operator installs at the customer premise.

To measure latencies in the core network and 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 and LMAP collector and controller at a more centralized location for example the network operations center which oversees multiple hub/headend locations or perhaps even the whole network of measurement agents. See Figure below.

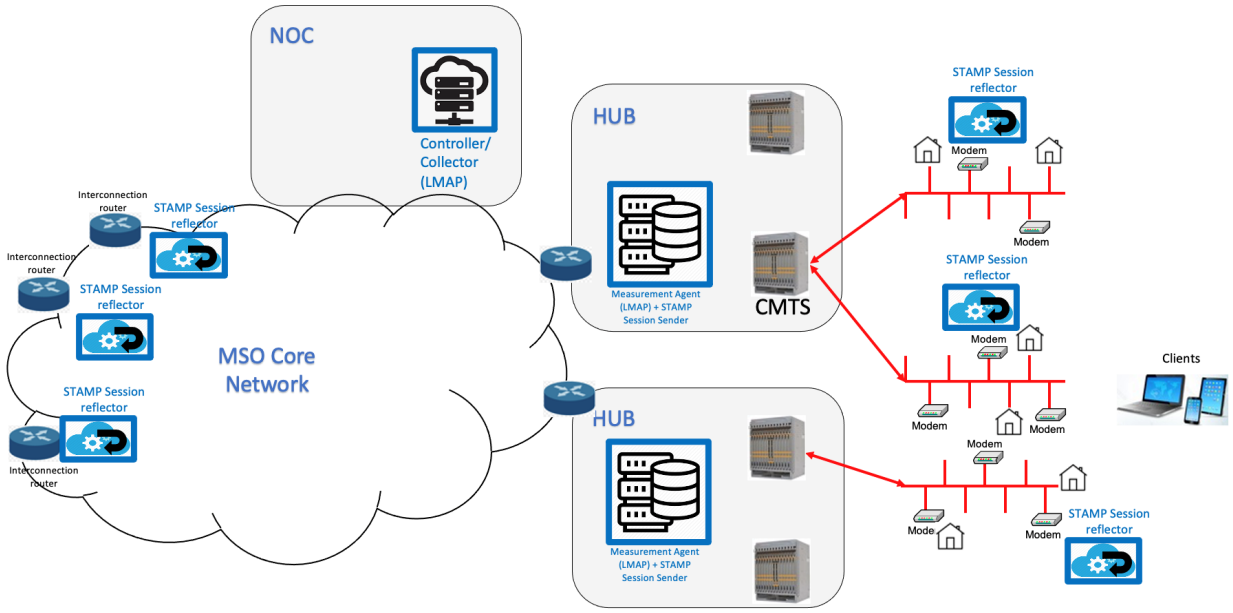

Figure 10 - Latency Measurement Architecture in a Cable Operator Network

4.2. Measurements in the Access vs Core vs Home Network

For the access network, each measurement agent at the hub locations would be 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 they can get a baseline understanding off the latencies across each of the potential paths across the core network.

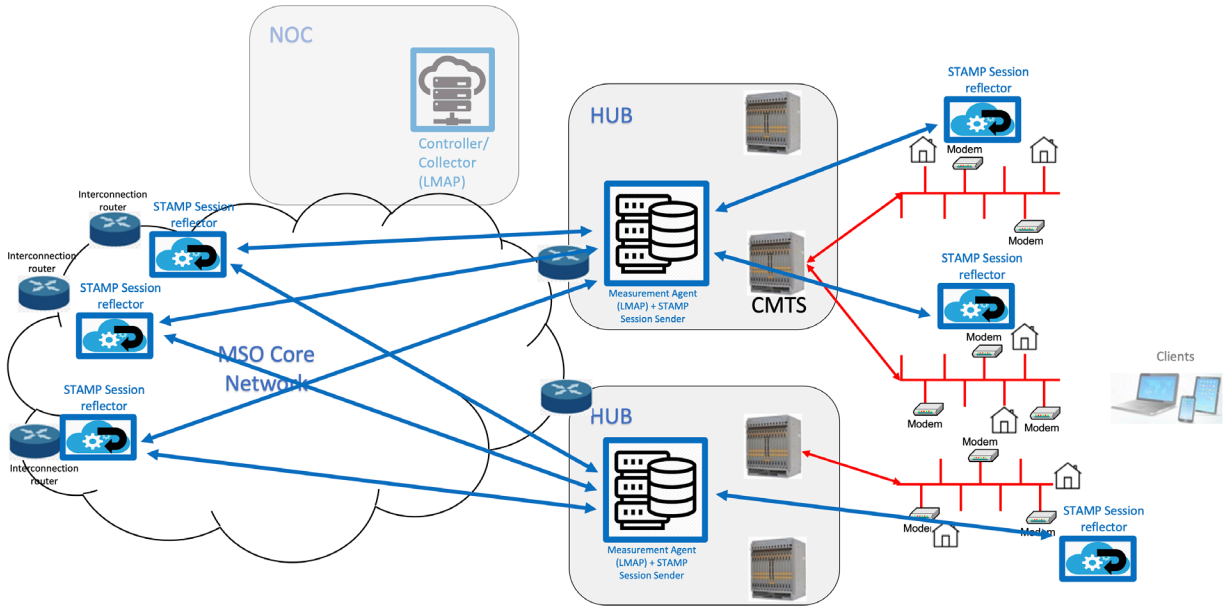

Figure 11 - STAMP Latency measurements (Access & core)

And operator can also instantiate a measurement agent within a client device for example 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 their customer premise, this will this 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.

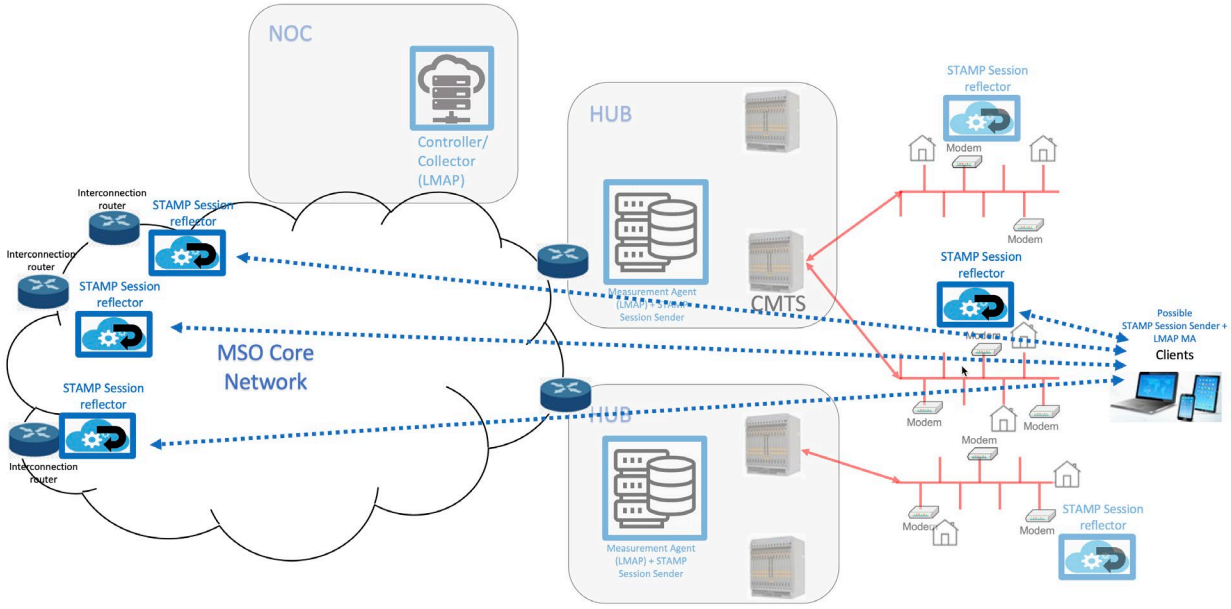

Figure 12 - STAMP Latency measurements from a Client-side MA

4.3. Measurement aggregation

Now for a large-scale measurement system the LMAP controller/collector will coordinate with each of the measurement agents in the network. The controller will command each of the measurement agents to run specific latency tests at a particular point in time or on a schedule. Each measurement agent will collect the set of latency measurements compute the requested statistics be it histogram counts or percentile data for that test. These results will be 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 the data analytics and data aggregation off the latency measurements across the network both the access network and the core network and potentially the home network will be implemented at the collector.

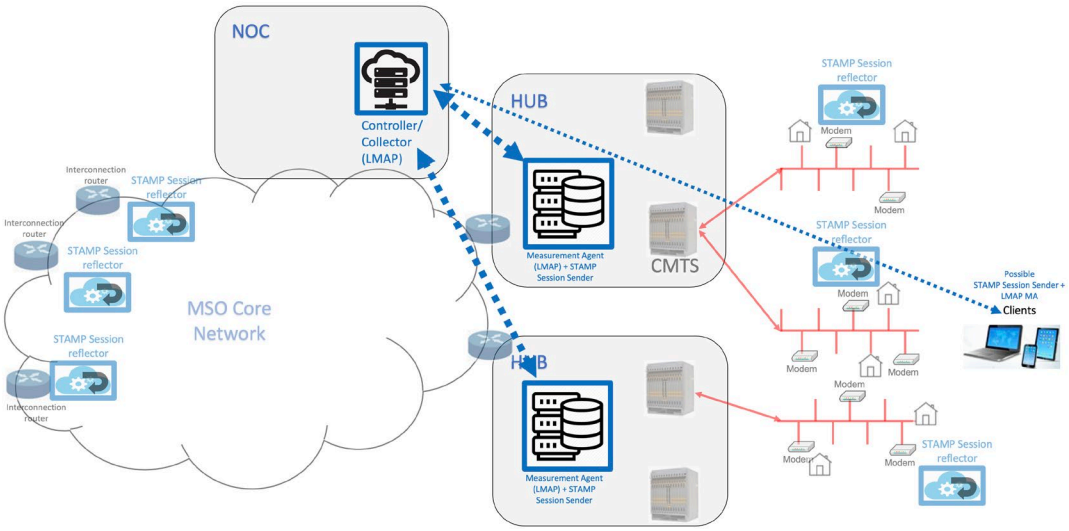

Figure 13 - LMAP Measurements control and reporting

4.4. Scaling Considerations

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

4.4.1. 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 network. In an informal survey we found that each operator has on the order of 8 to 12 peering points and transit locations and while of the bigger operators having up to 30 interconnect locations (based on past mergers and acquisitions of cable properties). Typical operator networks vary from 100s to 1000s of CMTSs and this implies 10s to 100s CMTS hub/headend locations.

An operator would desire to get a full mesh of latency measurement of "CMTS" x "Interconnect" locations. For a small cable operator this would be in the order of 100 links and up to a 1000 links for a larger operator.

An operator will likely place one Session reflector at each interconnect location. An operator may choose to place a one Measurement Agent at each hub/headend location. Alternatively, they could place to near the Core/Aggregation router to reduce number of measurement agents, those this will lose the ability to truly isolate access network latency in the measurements.

4.4.2. Access network Latency

As per the previous section, the assumption here is that an operator places measurement agents at each hub/headend location. Now the question is how many section reflectors an operator wants in the access portion of the network. An operator 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, would it be 20%, 10%, or 1%? For e.g., 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 their CMTS footprint, let's assume a 1000 CMTSs and each CMTS supporting 50 nodes, then this is just about 2 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 the coverage of the modems that are 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 RPD and RMD 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 then a phased approach to increasing measurement coverage across the network, will be the likely path than operators take.

5. Experimental results

To put all the latency measurement theory and protocols to test, we built an experimental latency measurement system prototype which was tested with measurement server and peer locations across the Internet.

5.1. 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 as well. These prototype components will be made available at [C3 CableLabs] after the development is complete.

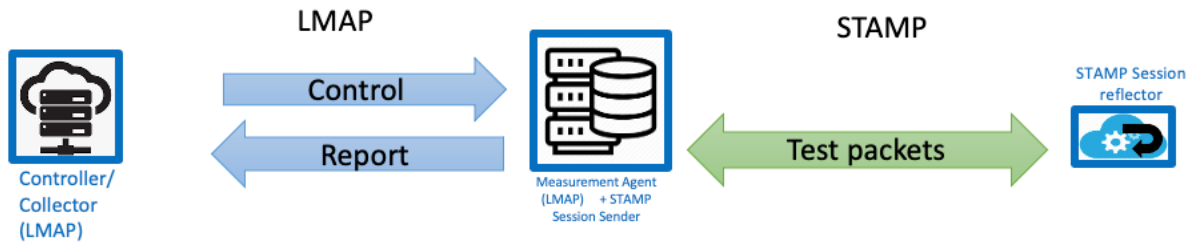

Figure 14 – Prototype components

5.1.1. Session-Reflector

The STAMP session-reflector software implementation was developed using C and 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 which could potentially be embedded within a cable modem or a gateway device in the house or a Wi-Fi AP. Another option would be a stand-alone device which the operator installs as "probes" throughout their network.

5.1.2. 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 built for a Unix environment. The Measurement Agents were instantiated on 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 here is to create a robust STAMP session sender software which 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 opensource server for implementing network configuration management based on the NETCONF Protocol. The server uses 'sysrepo' (an opensource 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.

5.1.3. 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 happens hereat the controller/collector.

5.2. Test Metrics

There are various metrics which and operator could track when they are looking to understand the Latency performance of their 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.

The [LM SCTE 20] paper discusses the basics of latency and proposes some metrics to measure. The main measures which an operator should look at are

- Latency RTT (Round trip time)
- PDV (Packet Delay Variation)
- 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, 100th percentile (maximum).

To understand how the latency varies over time and visualizing it, a simple time series graph shows a lot of interesting patterns. Additionally, a histogram of the data set 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.

5.3. Expermental Setup for Latency Measurement using STAMP

As a first step we decided to deploy the STAMP measurement agents and session reflectors in a network and work through the issues we would see with such a deployment on the Internet. Four different locations of the AWS servers were chosen, across the U.S, to give different sampling/variation of latency measurements. These server locations were in Oregon, N. California, Ohio, N. Virginia. There was a single session-reflector behind a DOCSIS 3.1 CM in a home in Denver. The session-reflector was connected via WiFi to the CM.

Figure 15 – Location of Measurement Agents & Session Reflector

5.4. Latency Measurement Test results

Each Measurement Agent ran latency tests to the Session reflector periodically. The interval chosen was every 5 mins, and tests were run over a period of a week. An individual 'test' consisted of 2000 to 5000 packets sent back-to-back.

Round trip latency and loss measurements were the focus of the experiments. The change in sequence numbers on the sender and received side also exposed packet loss in either direction.

As the clocks across the Session-sender and the Session-reflectors were not synchronized, one way latency measurements are not reported, though they were calculated.

5.4.1. Time Series view

The following figure shows a set of roundtrip latency measurements between a measurement agent in Oregon (STAMP Session sender) and a measurement peer (STAMP session reflector) in Denver. They're following figure shows about 75 hours of latency testing with 5000 test packets being sent every five minutes.

One can easily see that though the latency is around 35 ms for a lot of the time, there are some clear periodic larger spikes of latency up to 140 ms and then some even larger spikes of latency up to 2.5 seconds. The orange dots on the bottom of the graph indicate the number of packets lost during the test.

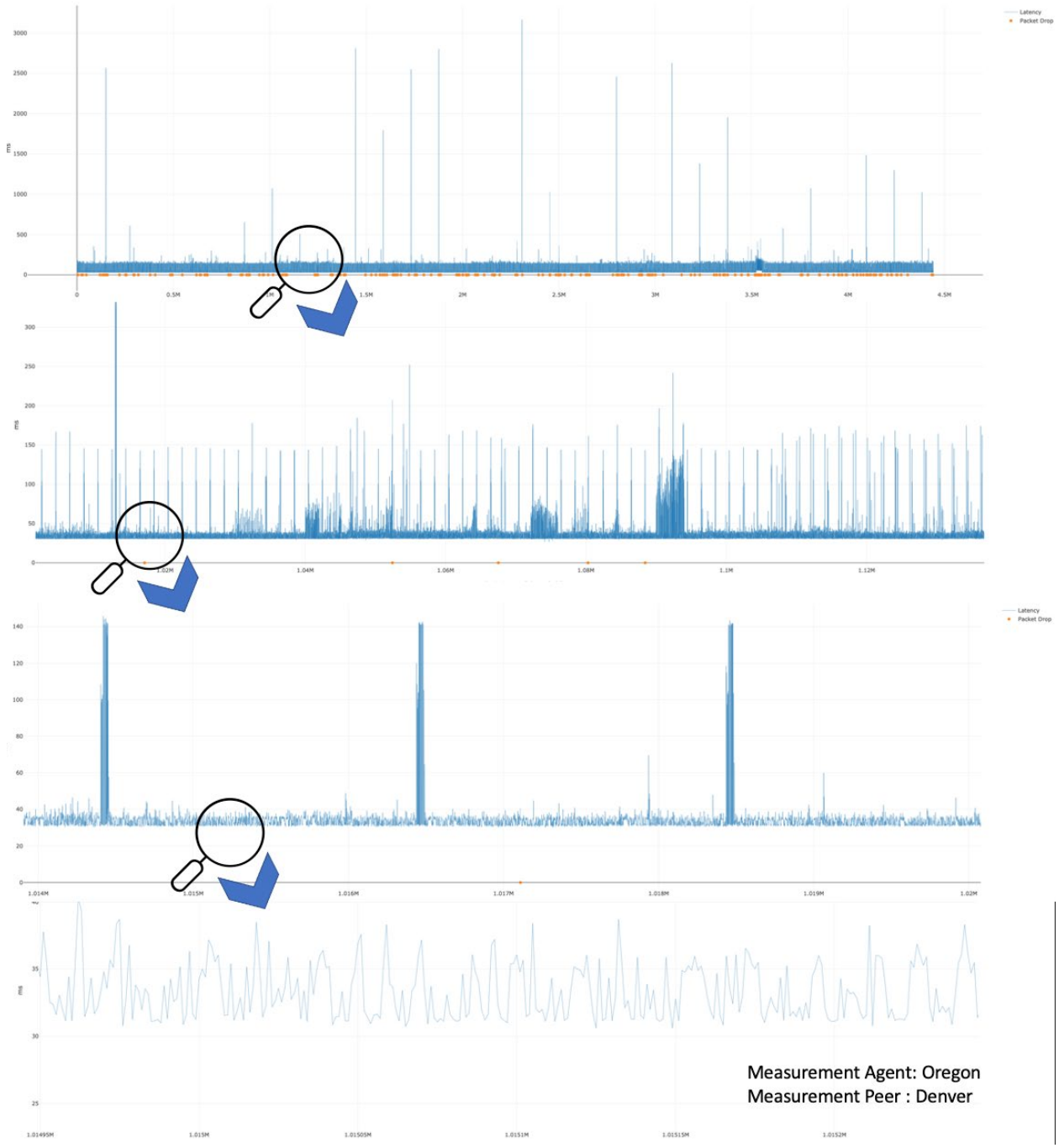

Figure 16 –Latency data - time series

5.4.2. Histogram view

The next figure below shows a histogram of latency, for each of the four links measured. The histogram bins are chosen to be one second each, do show with high granularity that different latency's that we observe to each of the measurement agents which are in different locations. As expected, the Ohio and N Virginia measurement agents have higher latencies (farther from Denver session reflector) compared to the measurement agents in California and Oregon (closer to Denver). As the histogram indicates, the

latency behavior is very different for different links and has multimodal distribution. Again, a reminder to network operators and users, to fight the urge to reduce latency to an "average" number or run a single test and take that to be latency number.

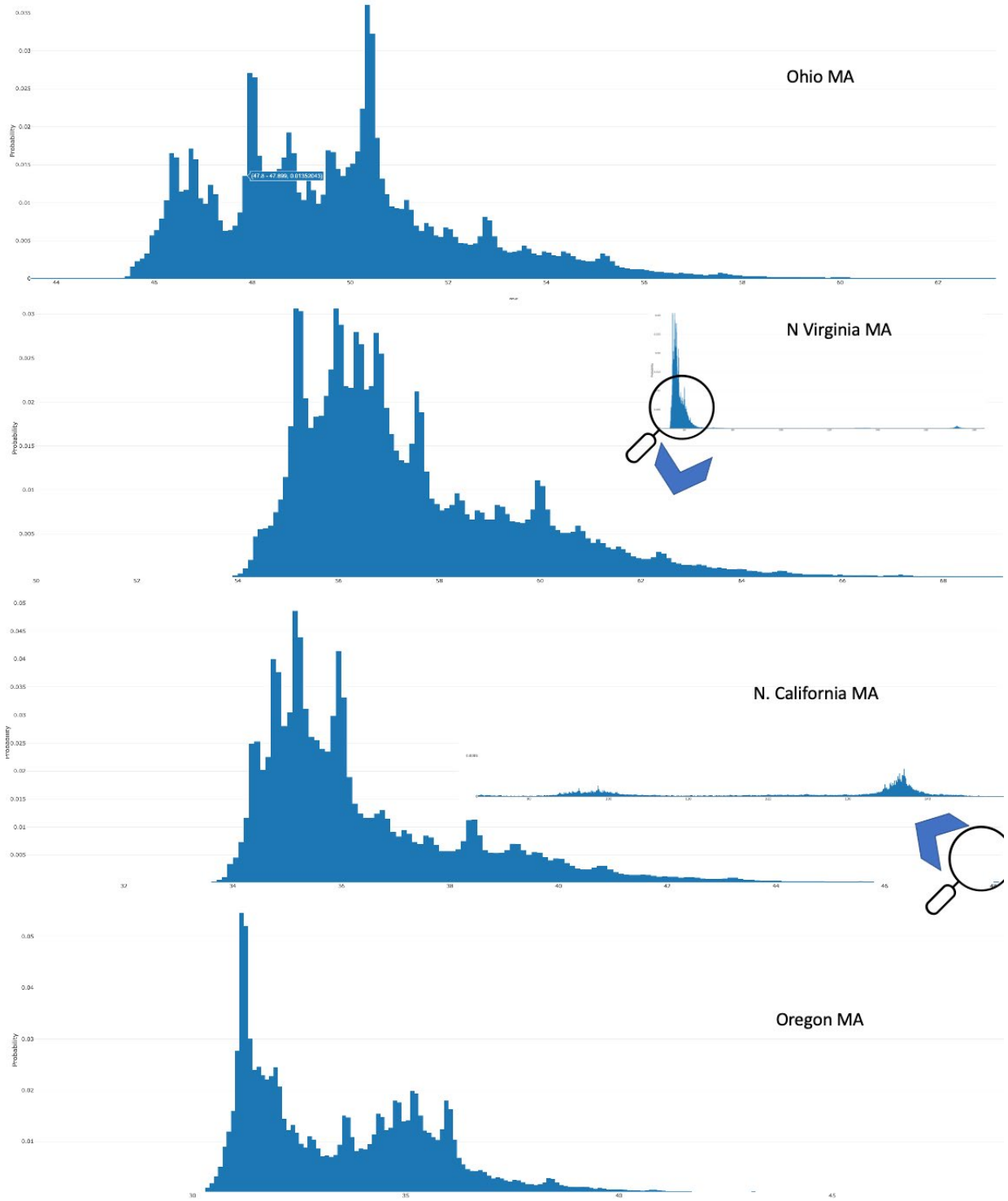

Figure 17 – Histogram of latencies from all 4 MAs

5.4.3. CDF view

The following figure shows a cumulative distribution function (CDF) graph of the latency data from each of the measurement agents. Here we look at the percentile values off Latency and use that as a metric to compare the different links. the scale on the Y-axis is logarithmic and the X-axis is zoomed in to the areas of interest. Operators are typically interested in the 99th percentile values of latency as that is shown to be indicative of the customer experience especially for real time applications such as gaming or real time communications. The table at the end of the section shows the 99th percentile, while the CDF graphs below show the 50th and the 95th percentile.

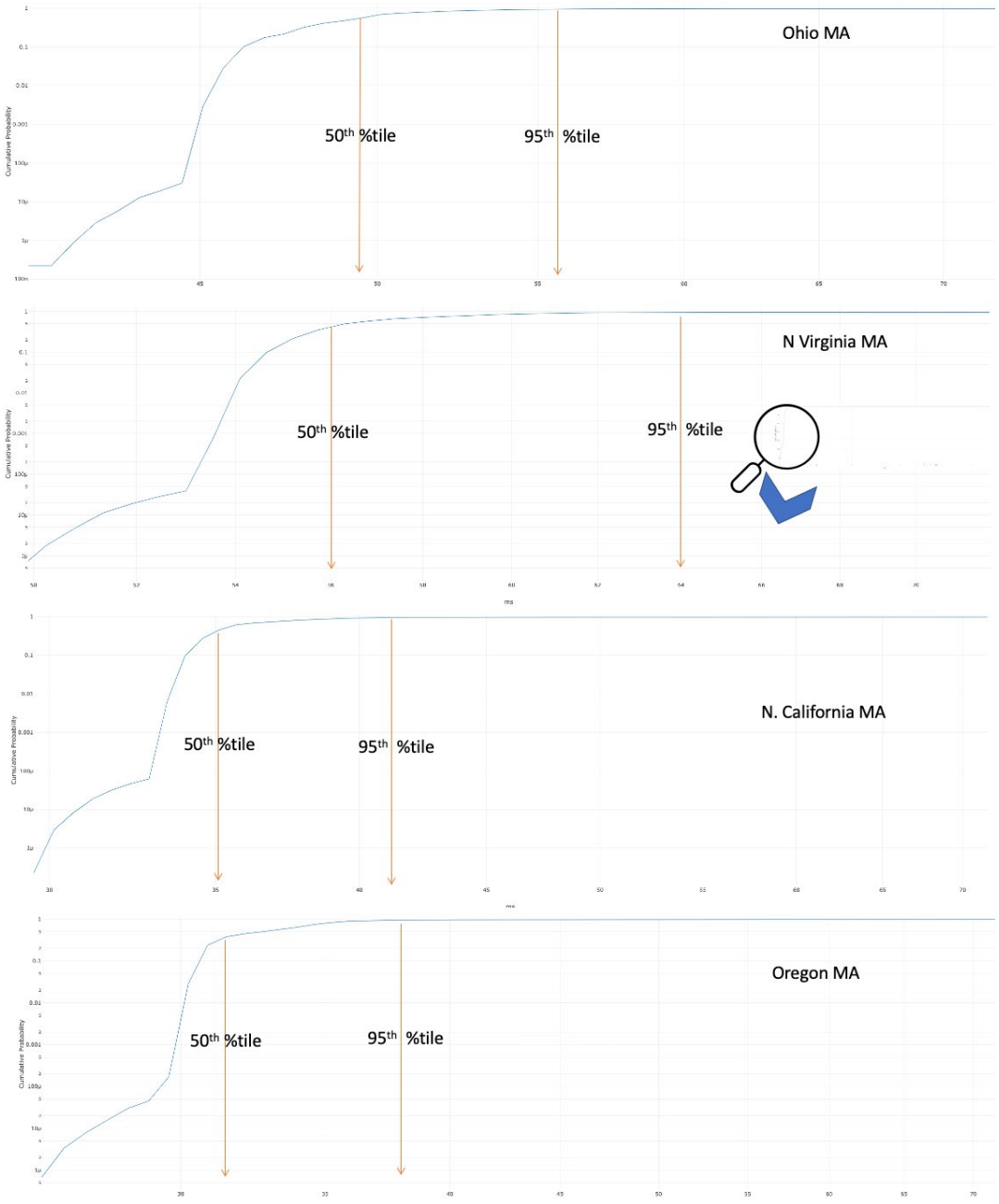

Figure 18 – CDF of latencies from all 4 MAs

5.4.4. Home Latency Testing

The next set of results below are for a measurement agent and a measurement peer located in the same home; this would be an example of testing the Wi-Fi latency within the home. the first graph below is a time series off a test run over 80 minutes, with 5000 packets being sent every five minutes. Again, while the nominal Latency hovers around the 5 ms mark, there are occasional spikes up to 14 to 16 ms and additional spikes of leading see up to 180 ms and some outliers of latency up to 1.2 seconds

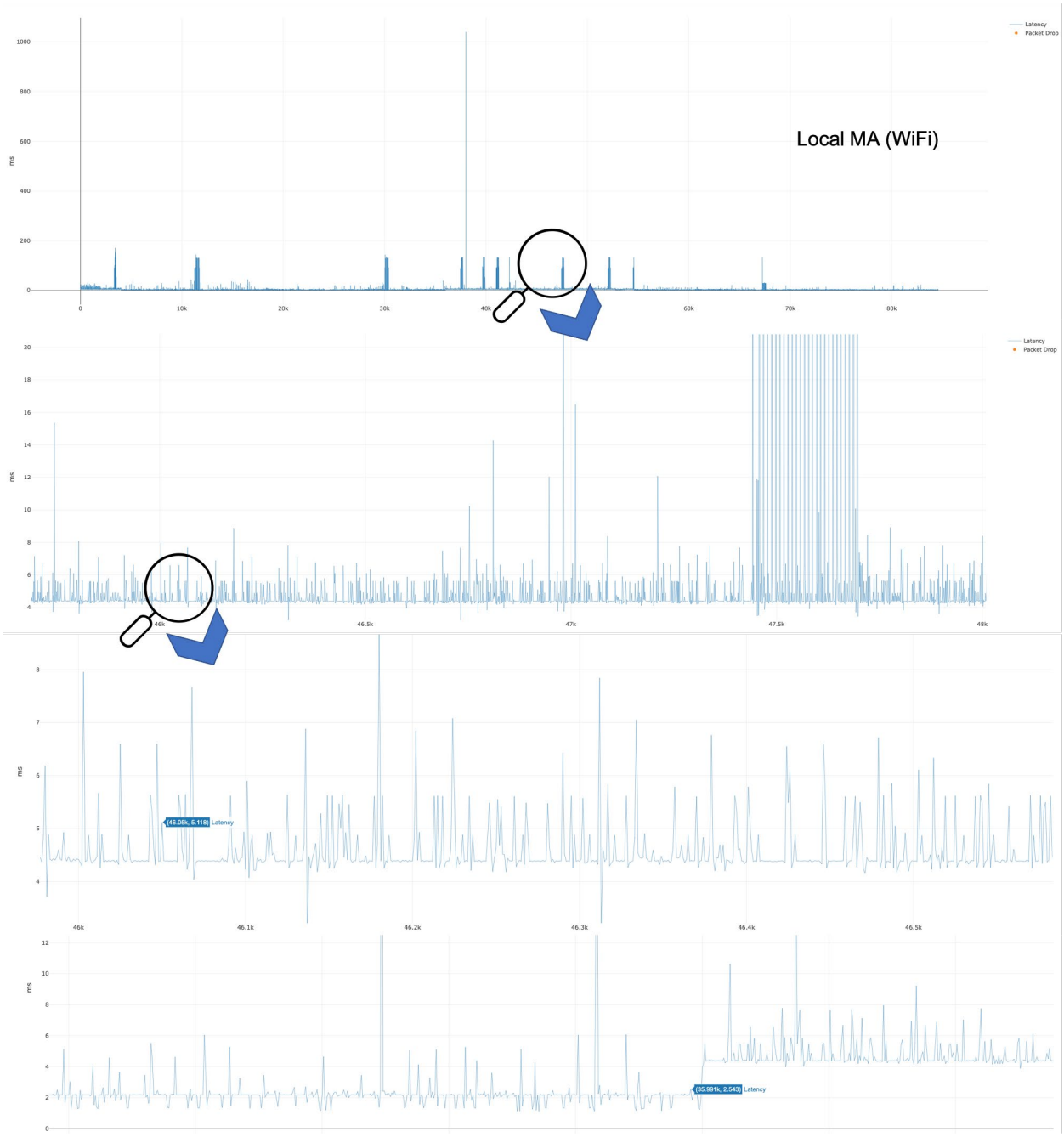

Figure 19 –LAN Wi-FI , Latency Time series

The LAN latency data, when distributed in two different bins and displayed as a histogram shows the 2.5 ms and 4.5 ms latency as the most common latencies observed in this test environment, and the CDF shows that the 50th percentile latency is around 2 milliseconds and the 95th percentile latency is around 5.3 milliseconds.

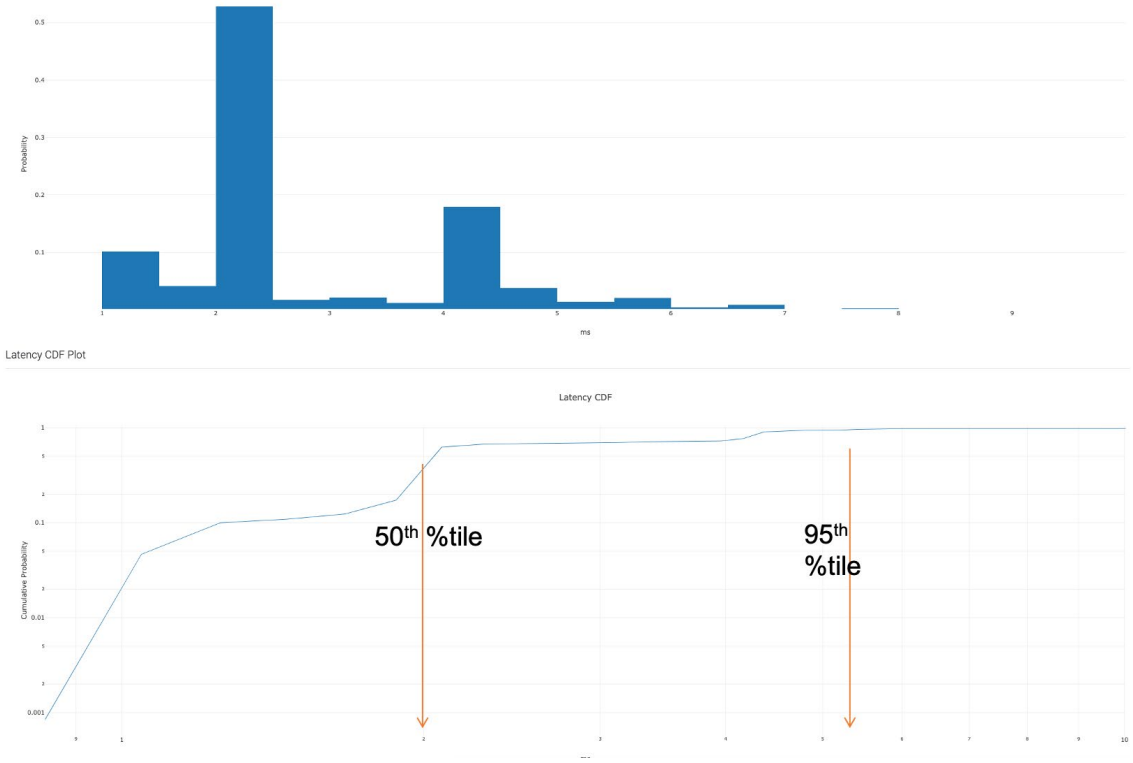

Figure 20 – LAN Histogram and CDF

5.4.5. Summary of results

The table below captures the Latency measurements percentiles of interest to the five different measurement agents four of which were spread across the country and the last was in the same LAN. Additionally, we also show the minimum and maximum latency observed across each of those five links.

Table 1 –Latency measurments to different locations

Server/Measurement Agent location (Measurement Peer in Denver)	**Min (0th percentile)**	**50th percentile**	**95th** percentile	**99th** percentile	**Max** (100th percentile)
N. Virginia	49.69 ms	56.3 ms	64.5 ms	172.2 ms	2.794 s
Ohio	40.63 ms	49.2 ms	56.6 ms	172.1 ms	2.812 s
California	29.56 ms	35.3 ms	42.5 ms	134.1 ms	2.808 s
Oregon	25.80 ms	32.7 ms	38.4 ms	116.8 ms	3.163 s
Local (LAN)	0.83 ms	2.02 ms	5.31 ms	13.41 ms	148.8 ms

6. Conclusion

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 needs planning and good engineering.

One main choice is that of a measurement protocol and here we successfully used STAMP across the Internet for latency measurement. STAMP is lightweight and easy to implement on existing hardware software platforms that it can be easily deployed by operators either in a standalone white box or as an add-on to existing devices (cable modems or gateways or APs). 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 big choice 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 percentiles values should give an operator very good understanding of latency performance of their networks and how it changes as they deploy newer technologies.

Abbreviations

PDV	packet delay variation
RTT	round trip time
CDF	cumulative distribution function
DSCP	Diff Serv Code Point
ms	millisecond
LMAP	Large-Scale Measurement of Broadband Performance
TWAMP	Two Way Active Measurement Protocol
STAMP	Simple Two-Way Active Measurement Protocol

Bibliography & References

[IETF RFC 8762] Simple Two-Way Active Measurement Protocol https://www.rfc-editor.org/rfc/rfc8762.html IETF, RFC 8762, 2020

[IETF RFC 8972] STAMP Optional Extensions https://datatracker.ietf.org/doc/html/rfc8972 , IETF RFC 8972, 2021

[IETF RFC 7594] A Framework for Large-Scale Measurement of Broadband Performance (LMAP) https://datatracker.ietf.org/doc/html/rfc7594 , IETF RFC 7594, 2015

[IETF RFC 8193] Information Model for Large-Scale Measurement Platforms (LMAPs) https://datatracker.ietf.org/doc/html/rfc8193 IETF, RFC 8193, 2017

[IETF RFC 8194] YANG Data Model for LMAP Measurement Agents, https://datatracker.ietf.org/doc/html/rfc8194 IETF, RFC 8194, 2017

[IETF RFC 7223] A YANG Data Model for Interface Management https://datatracker.ietf.org/doc/html/rfc7223 , IETF RFC

[LM SCTE 20] Latency Measurement: What is latency and how do we measure it? Karthik Sundaresan, Greg White, Steve Glennon, SCTE 2020

[C3 CableLabs] CableLabs Common code community, https://community.cablelabs.com/wiki/display/C3

