S69013_002

Draft Standard

MEF 66 Draft (R2)

SOAM for IP Services

Release 2

March 2019

This draft represents MEF work in progress and is subject to change.

This draft document represents MEF work in progress, has not achieved full MEF standardization and is subject to change. There are known unresolved issues that are likely to result in changes before this becomes a fully endorsed MEF Standard. The reader is strongly encouraged to review the Release Notes when making a decision on adoption. Additionally, because this document has not been adopted as a Final Specification in accordance with MEF's Bylaws, Members are not obligated to license patent claims that are essential to implementation of this document under MEF's Bylaws.

Disclaimer

Table of Contents

1 List of Contributing Members 1

2 Abstract 1

3 Release Notes 1

4 Terminology and Abbreviations 2

5 Compliance Levels 5

6 Numerical Prefix Conventions 5

7 Introduction 6

7.1 Document Structure 7
7.2 Use Cases 7

8 Fault Management 9

8.1 FM Use Cases 9
8.1.1 End-to-End Monitoring 9
8.1.2 UNI Access Link 11
8.1.3 IPVC Monitoring 13
8.2 FM Tool Requirements 14
8.2.1 Proactive Monitoring 14
8.2.1.1 BFD Overview 14
8.2.1.2 BFD Support 15
8.2.2 On-Demand Fault Monitoring 16
8.3 FM Reporting 18

9 Performance Management 20

9.1 PM Use Cases 20
9.1.1 Location to Location Monitoring 25
9.1.2 IPVC Monitoring 27
9.2 PM Common Requirements 30
9.2.1 Life Cycle 30
9.2.1.1 General Overview of Parameters 30
9.2.1.2 Proactive and On-Demand PM Sessions 31
9.2.1.3 Create 31
9.2.1.4 Delete 32
9.2.1.5 Start and Stop 32
9.2.1.6 Measurement Intervals 33
9.2.1.7 Repetition Time 34
9.2.1.8 Alignment of Measurement Intervals 34
9.2.1.9 Summary of Time Parameters 35
9.2.2 Storage 35
9.2.2.1 Measurement Interval Data Sets 37
9.2.2.2 Measurement Bins 38
9.2.2.3 Volatility 40
9.2.2.4 Measurement Interval Status 41
9.3 PM Implementation Requirements 41
9.3.1 PM Implementation Description 43
9.4 PM Tool Requirements 54

9.4.1 Active Measurement 54
9.4.1.1 TWAMP Light *54*
9.4.1.2 STAMP *55*
9.4.1.2.1 Session-Sender Behavior 55
9.4.1.2.2 Session-Reflector Behavior 55
9.4.1.2.3 Interoperability with TWAMP Light 56
9.4.1.3 TWAMP *56*
9.4.1.3.1 Session-Sender Behavior 56
9.4.1.3.2 Session-Reflector Behavior 57
9.5 Threshold Crossing Alerts (TCAs) 57
9.5.1 TCA Reporting 58
9.5.1.1 Stateless TCA Reporting *58*
9.5.1.2 Stateful TCA Reporting *59*
9.5.2 SOAM PM Thresholds for TCAs 60
9.5.3 SOAM PM TCA Notification Messages 67
10 Hybrid Measurement 69
10.1 Alternate Marking Explanation 69
10.1.1 Single-Marking Methodology 71
10.1.2 Mean Delay 71
10.1.3 Double-Marking Methodology 72
10.2 Alternate Marking for FM 72
10.3 Alternate Marking for PM 72
11 References 74
Appendix A Life Cycle Terminology (Informative) 77
A.1 Proactive PM Sessions 77
A.2 On-Demand PM Sessions 78
A.3 PM Sessions With Clock-Aligned Measurement Intervals and Repetition Time of "None" 79
A.4 PM Sessions With Clock-Aligned Measurement Intervals and Repetition Times Not Equal To "None" 80
Appendix B Measurement Bins (Informative) 84
B.1 Description of Measurement Bins 84
B.2 One-way Packet Delay Performance 85
B.3 One-way Inter Packet Delay Performance 85
B.4 One-way Packet Delay Range Performance 85
B.4.1 Case 1: $Q_1(x)$ 85
B.4.2 Case 2: $Q_h(x)$ 86
Appendix C Statistical Considerations for Loss Measurement (Informative) 87
C.1 Synthetic Packets and Statistical Methods 87
Appendix D Normalizing Measurements for PDR (Informative) 94
D.1 Topology Shifts 95
D.1.1 Minimum Delay Becomes Significantly Smaller 95
D.1.2 Minimum Delay Becomes Significantly Larger 95
D.2 Impact of Lack of ToD Synchronization 96

Appendix E Calculation of SLS Performance Metrics (Informative) 98
E.1 One-way Packet Delay 98
E.2 One-way Mean Packet Delay 99
E.3 One-way Packet Loss 99

List of Figures

Figure 1 – Example of an IPVC connecting three UNIs 8
Figure 2 – End-to-End BFD........ 10
Figure 3 – UNI Access Link BFD with Subscriber Provided CE........ 11
Figure 4 – UNI Access Link BFD with SP Managed CE........ 12
Figure 5 – PE-PE BFD Session 13
Figure 6 – SLS-RPs, MPs and Pair of MPs 21
Figure 7 – SLS Method 1 and Method 2 Comparison........ 23
Figure 8 - Example MP Locations........ 24
Figure 9 – Active PM Location to Location via IP-PMVC........ 26
Figure 10 – IPVC EP to IPVC EP Active Measurement........ 28
Figure 11 – Active Measurement when MPs are not at – IPVC EPs 29
Figure 12 – Example of Measurement Bins and Intervals........ 36
Figure 13 – Example of Packet Count Measurements........ 37
Figure 14 – Single-Ended Function........ 42
Figure 15 - Timestamp Locations........ 47
Figure 16 – Stateless TCA Reporting Example........ 59
Figure 17 – Stateful TCA Reporting Example 60
Figure 18 – Upper Bin Count for Threshold Crossing 61
Figure 19 – AltM description........ 70
Figure 20 – AltM measurement strategies........ 71
Figure 21 – Measurement Interval Terminology........ 78
Figure 22 – Illustration of non-Repetitive, On-Demand PM Session........ 79
Figure 23 – Example of Repetitive On-Demand PM Session 79
Figure 24 – Example Proactive PM Session with Clock-Aligned Measurement Interval........ 80
Figure 25 – Example On-Demand PM Session with Clock-Aligned Measurement Interval........ 81
Figure 26 – Second Example of On-Demand PM Session with Clock-Aligned Measurement Interval........ 82
Figure 27 – Hypothesis Test for Synthetic Packet Loss Measurements........ 87
Figure 28 – Density Curve and Probability of Exceeding the Objective........ 88
Figure 29 – Synthetic Loss Performance Example 1........ 89
Figure 30 – Synthetic Loss Performance Example 2........ 90
Figure 31 – Synthetic Loss Performance Example 3........ 90
Figure 32 – Synthetic Loss Performance Example 4........ 91
Figure 33 – Example PDR Distribution (normalized), and Bins........ 94
Figure 34 – Reduction in Minimum Delay, due to Network Topology Change 95
Figure 35 – Increase in Minimum Delay, due to Network Topology Change 96
Figure 36 – Lack of ToD Synchronization 96

List of Tables

Table 1 – Terminology and Abbreviations 4
Table 2 – Numerical Prefix Conventions 5
Table 3 - On-demand Tool Recommended Defaults 18
Table 4 – Time Parameters 35
Table 5 – Example Measurement Bin Configuration 40
Table 6 – Mandatory Stateful Single-Ended Data Set 51
Table 7 – Mandatory Stateless Single-Ended Data Set 53
Table 8 – Mandatory Single-Ended Data Set with Clock Synchronization 54
Table 9 – SOAM Performance Metrics TCA 63
Table 10 – TCA Notification Message Fields 68
Table 11 – Comparison of TCA Fields in X.73x and MEF 61 68
Table 12 – CoV Calculations with Message Period 1s 93
Table 13 – CoV Calculations with Message Period 100ms 93

1 List of Contributing Members

The following members of the MEF participated in the development of this document and have requested to be included in this list.

Editor Note 1: *This list will be finalized before Letter Ballot. Any member that comments in at least one CfC is eligible to be included by opting in before the Letter Ballot is initiated. Note it is the MEF member that is listed here (typically a company or organization), not their individual representatives.*

- ABC Networks
- XYZ Communications
- ACME Corporation

2 Abstract

This document specifies Service Operations, Administration, and Maintenance (SOAM) of IP Services described using the IP Service Attributes as defined in MEF 61.1 [33]. This covers both Fault Management (FM) and Performance Management (PM) of IP services.

The scope of this document is to define how Service Operations, Administration, and Maintenance (SOAM) Fault Management (FM) and Performance Monitoring (PM) can be applied to IP Services described using Service Attributes defined in MEF 61.1 [33]. The goal of this document is to define a set of specific fault and performance measurement methods that are recommended to be implemented by equipment providers and Service Providers. The methods defined include Proactive and On-demand Fault Management and active Performance Monitoring.

The focus of FM is on Bidirectional Forwarding Detection (BFD) as defined in RFC 5880 [11], RFC 5881 [12], and RFC 5883 [13] for Proactive monitoring. Ping and traceroute using ICMP as defined in RFC 792 [2] and RFC 4443 [8] are used for On-demand monitoring and defect localization. These tools are well defined and broadly implemented today. This document defines options, modes, and parameters for these tools based on defined use cases. The focus of PM for Active Measurement is on Two-Way Active Measurement Protocol (TWAMP) and TWAMP Light as defined in RFC 5357 [10] and Simple Two-way Active Measurement Protocol (STAMP) as defined in draft-ietf-ippm-stamp [20]. TWAMP, TWAMP Light, and STAMP are included in the scope to cover both complex and more simplified implementations.

3 Release Notes

There are no release notes for this Draft Standard.

4 Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions to terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling, in other MEF or external documents.

In addition, terms defined in MEF 61.1 [33] are included in this document by reference, and are not repeated in the table below.

Term	Definition	Reference
BFD	Bidirectional Forwarding Detection	IETF RFC 5880 [11]
Bidirectional Forwarding Detection	A protocol intended to detect faults in the bidirectional path between two forwarding engines, including interfaces, data link(s), and to the extent possible the forwarding engines themselves, with potentially very low latency.	IETF RFC 5880 [11]
ICM	Infrastructure Control and Management	MEF 55.1
ICMP	Internet Control Message Protocol	IETF RFC 792 [1] IETF RFC 4443 [8]
ICMP Ping	A common term for a tool that uses an ICMP Echo or Echo Reply Message as defined in RFC 792 [2] for IPv4 and RFC 4443 [8] for IPv6.	This document
Infrastructure Control and Management	The set of functionality providing domain specific network and topology view resource management capabilities including configuration, control and supervision of the network infrastructure. ICM is responsible for providing coordinated management across the network resources within a specific management and control domain. For example, a system supporting ICM capabilities provides connection management across a specific subnetwork domain. Such capabilities may be provided within systems such as subnetwork managers, SDN controllers, etc.	MEF 55 [32]
LSP	Label Switched Path	IETF RFC 3031 [5]
MD5	Message Digest Algorithm	IETF RFC 1321 [3]
Measurement Interval	A period of time during which measurements are taken. Measurements initiated during one Measurement Interval are kept separate from measurements taken during other Measurement Intervals.	MEF 35.1 [31]

Term	Definition	Reference
Measurement Point	An actively managed SOAM entity associated with a specific service instance that can generate and receive SOAM PDUs and track any responses.	This document
MI	Measurement Interval	MEF 35.1 [31]
MP	Measurement Point	This document
MPLS	Multi-Protocol Label Switching	IETF RFC 3031 [5]
On-demand	SOAM actions that are initiated via manual intervention for a limited time to carry out diagnostics.	MEF 35.1 [31]
Proactive monitoring	SOAM actions that are carried on continuously to permit timely reporting of fault and/or performance status.	MEF 35.1 [31]
Service Operation Administration and Maintenance	Service OAM addresses Fault Management and Performance Monitoring of services and devices used to implement services.	This document
Service Orchestration Functionality	The set of service management layer functionality supporting an agile framework to streamline and automate the service lifecycle in a sustainable fashion for coordinated management supporting design, fulfillment, control, testing, problem management, quality management, usage measurements, security management, analytics, and policy-based management capabilities providing coordinated end-to-end management and control of Layer 2 and Layer 3 Connectivity Services.	MEF 55 [32]
SHA1	Secure Hash Algorithm	IETF RFC 3174 [6]
SM	State Machine	This document
SOAM	Service Operation Administration and Maintenance	This document
SOF	Service Orchestration Functionality	MEF 55 [32]
STAMP	Simple Two-way Active Measurement Protocol	IETF Draft draft-ietf-ippm-stamp [20]
TCA	Threshold Crossing Alert	GR-253 [34]
ToD	Time of Day	MEF 35.1 [31]
ICMP Traceroute	A common term that refers to the ability to use the Echo and Time Exceeded messages defined in RFC 792 [2] for IPv4 and RFC 4443 [8] for IPv6 to determine the routing path from the source address to the destination address.	This document

Term	Definition	Reference
TWAMP	Two-way Active Measurement Protocol	IETF RFC 5357 [10]
TWAMP Light	TWAMP Light is significantly simplified mode of TWAMP-Test part of TWAMP.	IETF RFC 5357, Appendix I [10]
UBC	Upper Bin Count	MEF 35.1 [31]
UTC	Coordinated Universal Time	ISO 8601 [23]

Table 1 – Terminology and Abbreviations

5 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119 [4], RFC 8174 [16]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as **[Ox]** for optional**.**

A paragraph preceded by **[CRa]<** specifies a conditional mandatory requirement that **MUST** be followed if the condition(s) following the “<” have been met. For example, “**[CR1]<**[D38]” indicates that Conditional Mandatory Requirement 1 must be followed if Desirable Requirement 38 has been met. A paragraph preceded by **[CDb]<** specifies a Conditional Desirable Requirement that **SHOULD** be followed if the condition(s) following the “<” have been met. A paragraph preceded by **[COc]<** specifies a Conditional Optional Requirement that **MAY** be followed if the condition(s) following the “<” have been met.

6 Numerical Prefix Conventions

This document uses the prefix notation to indicate multiplier values as shown in Table 2.

Decimal		**Binary**	
Symbol	**Value**	**Symbol**	**Value**
k	10^{3}	Ki	2^{10}
M	10^{6}	Mi	2^{20}
G	10^{9}	Gi	2^{30}
T	10^{12}	Ti	2^{40}
P	10^{15}	Pi	2^{50}
E	10^{18}	Ei	2^{60}
Z	10^{21}	Zi	2^{70}
Y	10^{24}	Yi	2^{80}

Table 2 – Numerical Prefix Conventions

7 Introduction

SOAM provides the protocols, mechanisms, and procedures for monitoring faults and the performance of an IP Virtual Connection (IPVC). The use of SOAM in IP Services is not standardized although IP Services are widespread. This document describes the tools that are needed, allowing equipment providers to understand what features and functions to include in their equipment, and provides recommendations to IP Service Providers (SP) on how to use these tools.

The document is divided into several sections covering Fault Management, Performance Management, and Hybrid Measurement. The Fault Management section includes Use Cases, FM Tool requirements, and FM reporting. The Performance Management section includes Use Cases, PM requirements, PM Tool requirements, and PM reporting. The Hybrid Measurement section includes informative discussion of Alternate Marking used for Hybrid Measurement. These sections reference previous MEF work, other Standards Bodies work, or might expand upon that work to support IP services.

For FM, Proactive monitoring and On-demand monitoring are specified. Proactive monitoring is defined as SOAM actions that are carried on continuously to permit timely reporting of fault and/or performance status. Within this document, Bidirectional Forwarding Detection (BFD) is specified as the tool to be used for Proactive fault monitoring. Recommendations for BFD options are included. On-demand fault monitoring is used to isolate a fault when one has been detected by Proactive monitoring or as a replacement for Proactive monitoring.

On-demand monitoring is defined as SOAM actions that are initiated via manual intervention for a limited time to carry out diagnostics. Ping and traceroute are the tools used for On-demand fault monitoring. Transmission and reception of ping and traceroute can use ICMP. Recommendations for options for these are included in this document.

For PM, Active Measurement using TWAMP Light/STAMP/TWAMP is specified. 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. MEF 61.1 [33] specifies one-way performance metrics which require Time of Day (ToD) clock synchronization for PD measurements. Since ToD clock synchronization is often difficult to implement, two-way measurements, divided in half and identified as derived measurements can be acceptable. Options for TWAMP, TWAMP Light, and STAMP are specified within the document. One Way Active Measurement Protocol (OWAMP) as defined in RFC 4656 [9] is not included in the scope of this document and is not recommended for use to perform PM due to the requirement to implement the control protocol at each end of the service.

Passive Measurement depends solely on observation of one or more existing packet streams. The streams are only used for measurement when they are observed for that purpose, but are present whether or not measurements take place. Passive Measurement is not within the scope of this document.

A Hybrid Measurement method is a combination of Active and Passive Measurement which makes observations on a dedicated measurement stream using header or marked bits included with an existing stream. The requirements for Hybrid Measurements are not discussed in this document. However, Section 10 describes one example of the Hybrid method, Alternate Marking. Hybrid Measurement methods such as Alternate Marking (AltM) are in the process of being defined. As other SDOs complete work on these methods, this document can be updated to include them.

7.1 Document Structure

This document is structured by measurement type. The Fault Management section contains use cases, tool requirements, implementation recommendations, and reporting requirements. The Performance Management section contains use cases, PM Solution requirements, Common PM Requirements, Storage Requirements, Threshold Crossing Alert Requirements, PM Tool requirements, implementation recommendations, and reporting requirements. The Hybrid Monitoring section provides an overview of AltM. Various appendices are provided to further assist with tool and implementation decisions.

7.2 Use Cases

The use cases shown in this document provide examples of how FM (section 8.1), PM (section 9.1), and AltM (section 10) can be used in a SPs network. These use cases are not all encompassing. Understanding how and why the SOAM tools are used will assist in understanding the requirements and recommendations that are provided in this document.

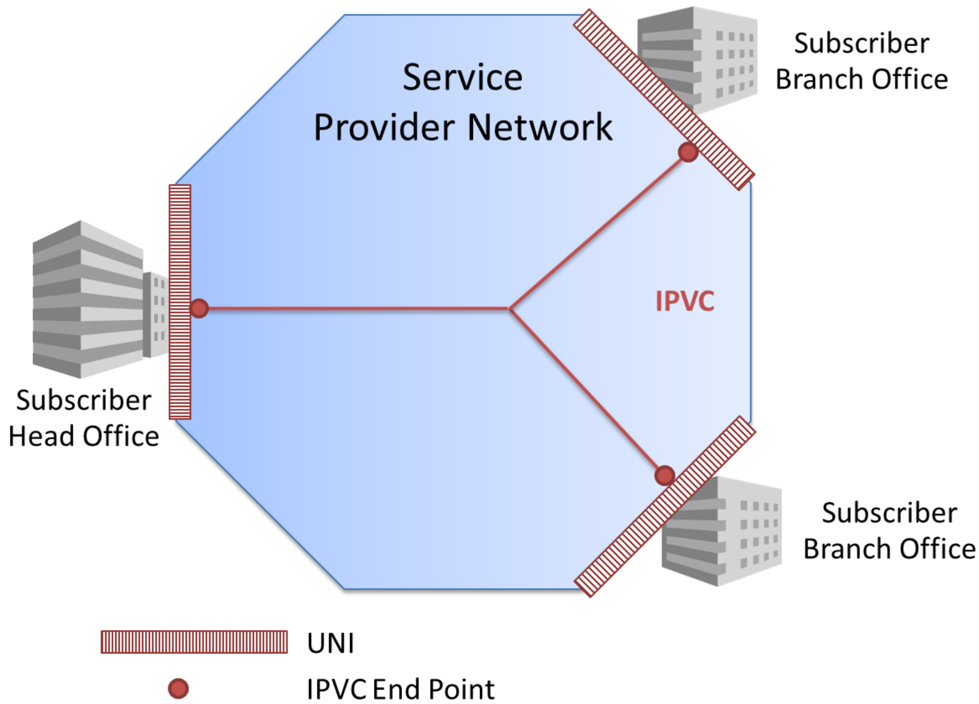

Figure 1 – Example of an IPVC connecting three UNIs

Figure 1 shows a basic IPVC. For the purposes of this document, this basic IPVC will be discussed in the use cases within this document. The single IPVC represented in Figure 1 connects three Subscriber locations. The SP desires to monitor faults and performance of this IPVC. The use cases within this document are used as examples and are provided as information only.

8 Fault Management

Fault Management (FM) provides the ability to detect failures within IP Services. This section contains the Use Cases, Tool Requirements, and Implementation Recommendations for FM for IP Services.

8.1 FM Use Cases

Faults that impact IP services include loss of connectivity due to network events, routing issues, equipment failures or other events. A fault is characterized as failure to pass packets as opposed to a performance degradation where packets can still pass but with excessive loss or delay. As mentioned previously in this document, BFD is the recommended tool for Proactive FM. BFD is a mature protocol that is widely implemented in CEs and PEs. For more information on BFD see section 8.2.1.

BFD is often used to detect faults on a single hop within a network. The use of BFD across a single physical link is out of scope except where used to detect faults on a UNI Access Link that is a single hop.

To support On-demand FM, tools such as ICMP Ping and ICMP Traceroute are used. These tools allow localization and isolation of a fault to be performed as needed. For more information on these tools see section8.2.2.

There are several ways that FM can be used to support IP services. Examples of these are shown in the following sections.

8.1.1 End-to-End Monitoring

An example of monitoring from IPVC End Point to IPVC End Point is shown in Figure 2. In this case, the SP demarcation equipment (CE) at the customer premises supports BFD, which is configured to run between each of the BFD Implementation (BFD IMP) at some regular interval.

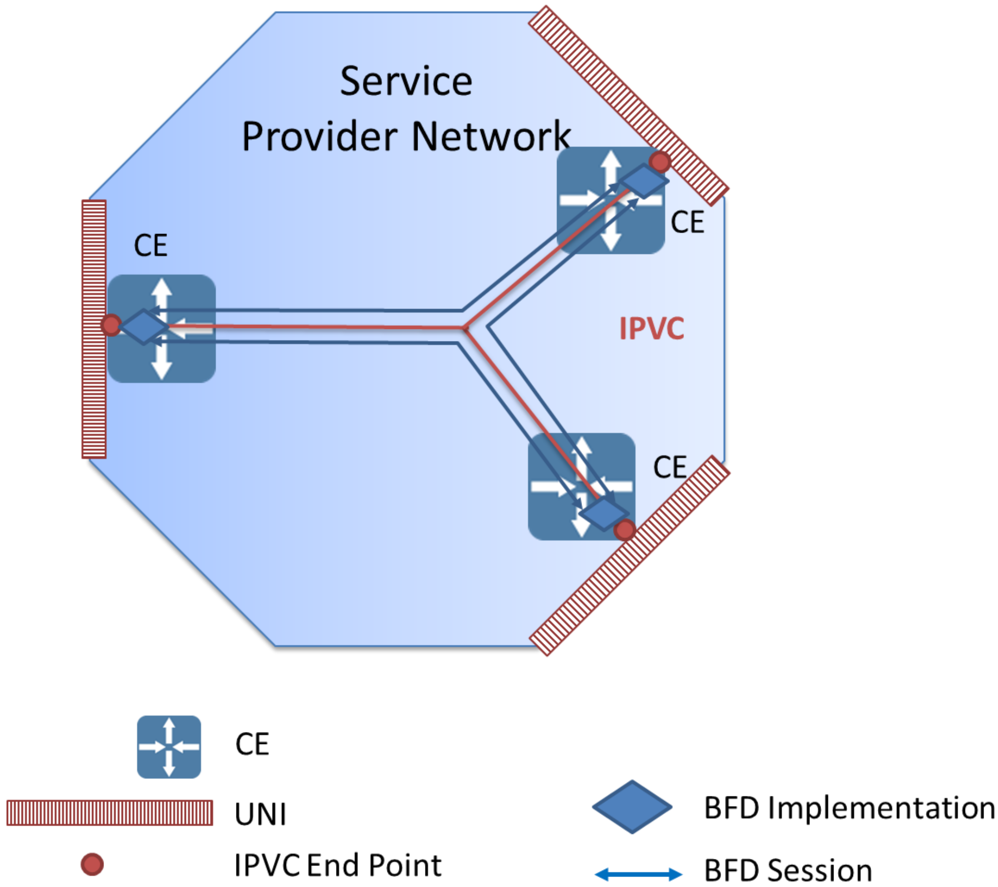

Figure 2 – End-to-End BFD

Figure 2 shows an Asynchronous BFD session between each of the IPVC End Points. Any failures of connectivity across the IPVC are detected. Examples of failures include loss of connectivity that occur between two IPVC EPs, high packet loss between two IPVC EPs that results in loss of contiguous BFD packets, or a fault in the CE that causes the BFD implementation to fail at an IPVC EP. Once the CEs are notified that a fault has occurred, they can take corrective action to reroute the packets to an alternate path. Depending on the transmission interval of BFD packets, fault detection can occur faster than routing protocol fault detection. The SP is able to configure a BFD session between the pair of CEs because the CEs are Provider-Managed. In the case of Subscriber-Managed CE, the SP is not able to configure a BFD session between the pair of CEs.

8.1.2 UNI Access Link

BFD can be configured to run between the Subscriber's CE and the SP's PE or between a SP managed CE and other Subscriber equipment across the UNI Access Link. MEF 61.1 [33] defines the UNI Access Link BFD Service Attribute which is used to define the BFD session attributes. In this case, BFD is being used to detect faults that occur on the UNI Access Link versus the CE to CE connectivity as discussed in section 8.1.1.

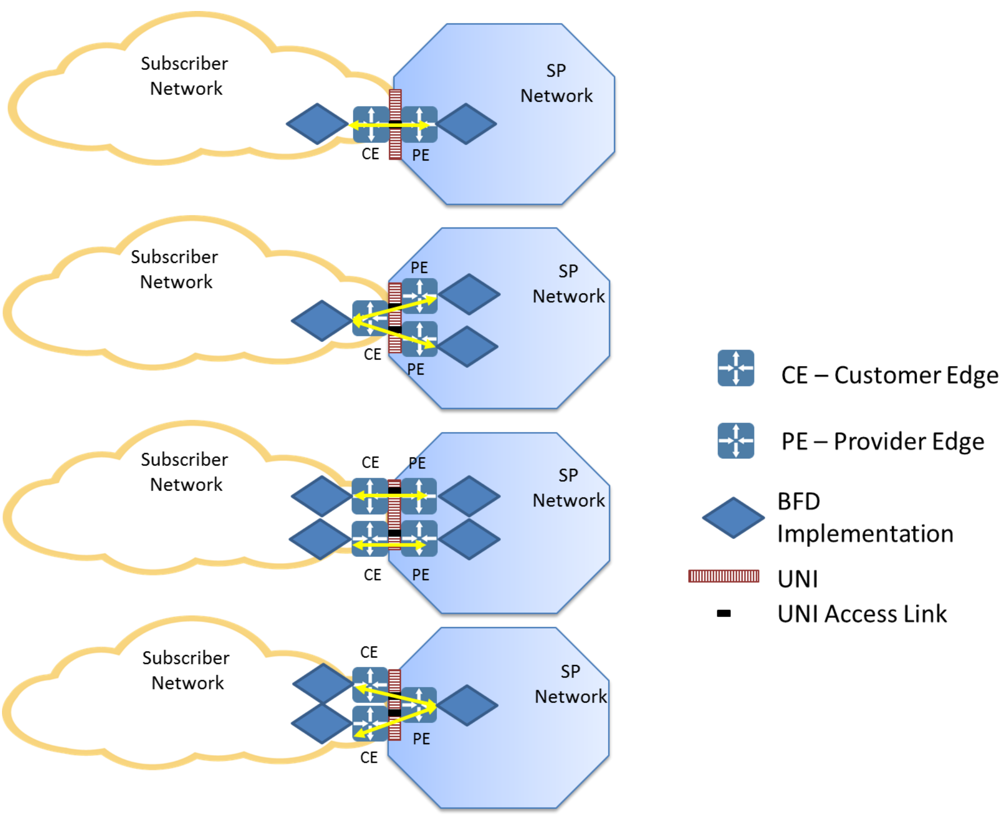

Figure 3 – UNI Access Link BFD with Subscriber Provided CE

Figure 3 shows several different UNI Access Link configurations when the CE is Subscriber-Managed. BFD sessions between the CE and the PE are configured and are used to detect faults on the UNI Access Link.

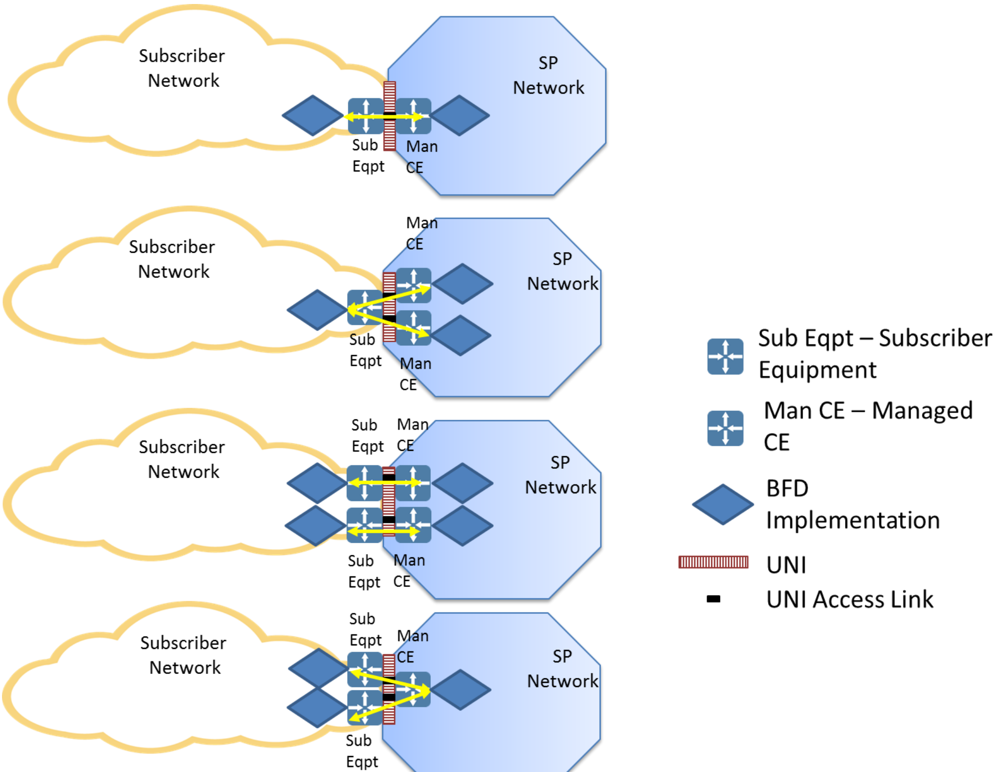

Figure 4 – UNI Access Link BFD with SP Managed CE

Figure 4 shows similar UNI Access Link configurations but in these configurations the CE is Provider-Managed. The BFD session is configured between the managed CE and some Subscriber equipment on the other side of the UNI Access Link.

Using BFD to monitor the UNI Access Link can be required if the physical connection between the CE and PE does not provide fault notification. The connection appears as a single hop and BFD is implemented as described in IETF RFC 5881 [12].

A BFD session that is active on the UNI Access Link can be used to detect faults that cause a rerouting of the Subscriber's traffic to another UNI Access Link. Such re-routing can occur only when there is an additional UNI Access Link that is not impacted by the fault.

Faults detected by the BFD session(s) in these Use Cases can include UNI interface failures, UNI physical connectivity failure, or CE, PE, or Subscriber Equipment failure.

8.1.3 IPVC Monitoring

When SPs do not provide the CE, they can still monitor an IPVC for faults. What they monitor might be a segment of the IPVC rather than the entire IPVC. In this example, the SP is using BFD between PE1 and PE2 to monitor a segment of the IPVC between PE1 and PE2.

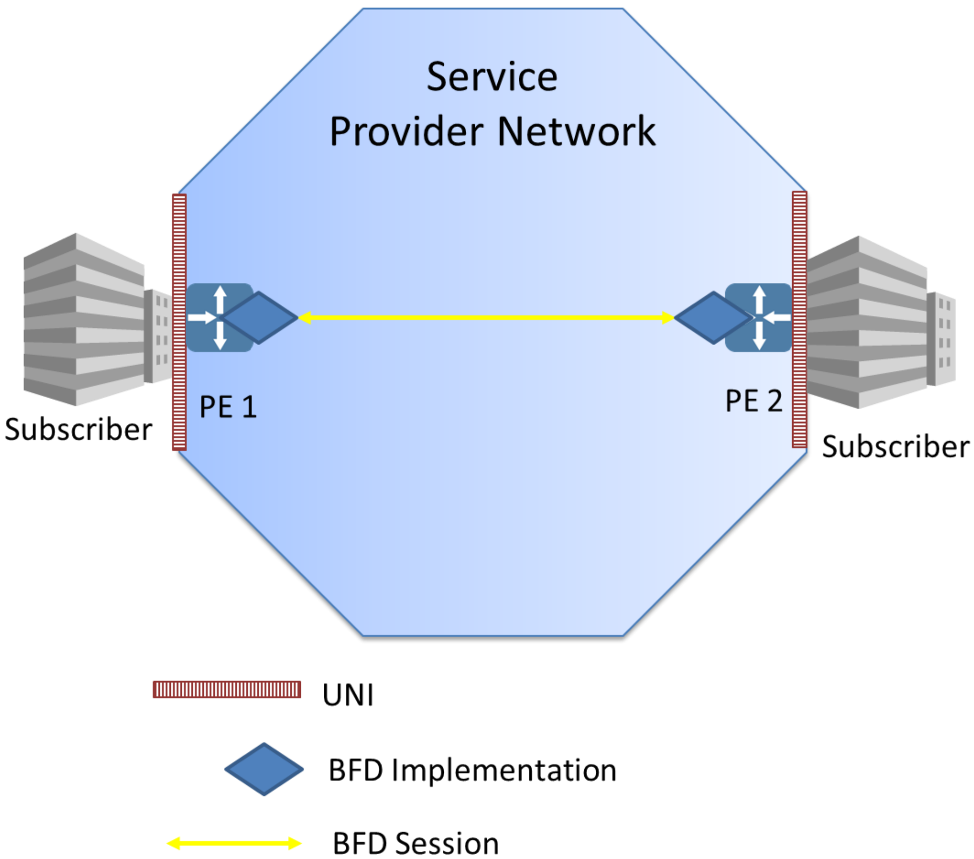

Figure 5 – PE-PE BFD Session

Figure 5 reflects an SP that is monitoring an IPVC from PE to PE. In some configurations the SP does not have any equipment at the Subscriber location. The SP uses BFD to monitor an IPVC from PE to PE since this is the most complete view of the service that they have. BFD is provisioned over the IPVC between the PEs, BFD control packets are exchanged, and IPVC loss of continuity between the PEs is detected. Examples of failures that can be detected include a loss of connectivity between PEs, a failure to reconverge after a failure, or a failure in a PE. BFD can detect faults faster than typical routing protocols and BFD can trigger routing protocols to reconverge reestablishing connectivity. To reconverge at least two paths need to exist between the PEs. If the SP has other protection mechanisms at lower levels, the BFD timer intervals need to take into account protection mechanism timers at these lower levels to ensure that the lower levels act before the BFD timer triggers a reconvergence.

8.2 FM Tool Requirements

As stated previously, BFD is being specified as the primary Proactive FM tool. ICMP ping and traceroute are specified as On-demand tools. This section of the document specifies the requirements that must be supported for each of these tool sets.

8.2.1 Proactive Monitoring

BFD is specified in IETF RFC 5880 [11]. Additional details on BFD intervals are specified in IETF RFC 7419 [14]. See RFC 5880 [11] for a detailed description of the BFD protocol and its operation. When proactively monitoring a single hop, BFD is implemented as described in RFC 5881 [12]. When proactively monitoring multihop services, BFD is implemented as described in RFC 5883 [13].

8.2.1.1 BFD Overview

Per RFC 5880 [11] BFD is intended to detect faults in the bidirectional path between two forwarding engines, including interfaces, data link(s), and to the extent possible the forwarding engines themselves, with potentially very low latency. BFD is a more efficient method to quickly detect and notify registered protocols that a failure has occurred. This means individual control protocols "hello" timers need not be configured individually and aggressively. They can rely on BFD for failure notification.

BFD operates between a pair of systems that are exchanging BFD packets. If a system stops receiving BFD packets for some specified period of time, the path is declared failed. A path is only declared up when properly constructed BFD packets are received at each system in the pair.

The time interval between the transmission of two consecutive BFD packets is negotiated between the two BFD systems. Because of random jitter of BFD packet transmission, average interval between two packets equals 0.875 of the negotiated value. RFC 7419 [14] provides recommendations on time intervals that are supported by all systems to make the negotiation process easier. Once the time interval is determined, RFC 5880 [11] defines two modes for BFD, Asynchronous and Demand. For FM Proactive monitoring, this document focuses on Asynchronous. Asynchronous mode provides a more proactive solution for monitoring for faults than Demand mode and can provide faster fault detection that a Demand session with the same transmission interval. The Echo function is an adjunct to both modes and allows one system to transmit BFD packets and the other systems loops them back through its forwarding path. While this can reduce the processing requirements to one end, it does add additional packets to the network.

Note: Echo function cannot be used with mulithop BFD specified in RFC 5883 [13].

Authentication can be supported by BFD to limit the ability of false packets to impact the forwarding paths. Authentication methods range from a simple password to MD5 and SHA1 authentication.

8.2.1.2 BFD Support

This section details requirements for network elements supporting BFD. BFD is defined in RFC 5880 [11]. RFC 5881 [12] and RFC 5883 [13] also apply for some implementations. Where support for a RFC is mandated, unless otherwise stated, all required and recommended requirements apply as stated in the RFC.

[R1] A BFD Implementation **MUST** comply with RFC 5880 [11] if BFD is supported.

[R2] A BFD Implementation **MUST** comply with RFC 5881 [12] if single hop BFD is supported.

[R3] A BFD Implementation **MUST** comply with RFC 5883 [13] if multi-hop BFD is supported.

Support for Demand mode, as specified in RFC 5880 section 6.6 [11], is optional. RFC 5880 [11] section 6.8.15 describes how the BFD implementation responds to a forwarding plane reset.

RFC 7419 [14] describes issues with negotiating BFD transmission intervals. To resolve these issues, it specifies a minimum list of common intervals that are to be supported.

[R4] A BFD implementation **MUST** support the following common intervals, 100ms, and 1 second as specified in RFC 7419 [14].

[D1] Other intervals specified in RFC 7419 [14], 3.3ms, 10ms, 20ms, 50ms, 10 seconds **SHOULD** be supported.

[R5] A BFD implementation **MUST** support a Detect multiplier of 3.

[D2] A BFD implementation **SHOULD** support a Detect multiplier range of 2-255

[R6] A BFD implementation that supports an interval in the list of 3.3ms, 10ms, 20ms, and 50ms **MUST** support all longer intervals in that list as specified in RFC 7419 [14].

Additional BFD transmission intervals can be supported.

[R7] An IP SOAM Implementation **MUST** support a mechanism to limit the number of IP SOAM FM packets processed per second.

As described previously a BFD implementation can be used to monitor either the Service Provider's network or services provided by the Service Provider for faults. Each of these might require that the IP Data Service packets containing the BFD packets be treated differently by the network devices. For this reason, the ability to set the DSCP value of the IP Data Service packets is required. The Service Provider might want to match the value of a Subscriber's service and use a different value for their network. The following requirements support these features.

[R8] An IP SOAM Implementation **MUST** support the ability to set the DSCP value of IP Data Service packets containing BFD packets.

[R9] The default value for the DSCP value **MUST** be 48.

8.2.2 On-Demand Fault Monitoring

On-demand fault monitoring uses Internet Control Message Protocol (ICMP) ping and traceroute. ICMP ping and traceroute use functions that are defined in RFC 792 [1] for IPv4 and RFC 4443 [8] for IPv6.

On-demand Fault Management for IPv4 is done using the Echo/Echo Reply and Time Exceeded messages defined in IETF RFC 792 [1]. This RFC defines widely deployed ICMP messages and header formats. On-demand Fault Management for IPv6 is done using the Echo Request/Echo Reply and Time Exceeded messages defined in IETF RFC 4443 [8]. This RFC defines widely deployed ICMP messages and header formats.

[R10] An On-demand Fault Monitoring implementation supporting IPv4 **MUST** comply with the requirements and message formats for Echo Request, Echo Reply, and Time Exceeded Messages as specified in RFC 792.

[R11] An On-demand Fault Monitoring implementation supporting IPv4 **MUST** support a unicast DA.

[R12] An On-demand Fault Monitoring implementation supporting IPv4 **MUST NOT** support a multicast DA.

[R13] An On-demand Fault Monitoring implementation supporting IPv6 **MUST** comply with the requirements and message formats for Echo Request, Echo Reply and Time Exceeded Messages as specified in RFC 4443.

[R14] An On-demand Fault Monitoring implementation supporting IPv6 **MUST** support a unicast DA.

[R15] An On-demand Fault Monitoring implementation supporting IPv6 **MUST NOT** support a multicast DA.

[R16] An On-demand Fault Monitoring implementation of ping **MUST** support a time interval between the transmissions of Echo Request messages of 1 second.

[D3] An On-demand Fault Monitoring implementation of ping **SHOULD** support a time interval between the transmissions of Echo Request messages of 100ms.

[R17] An On-demand Fault Monitoring implementation of ping **MUST** allow the number of Echo Request messages to be transmitted to be selected by the user.

[R18] An On-demand Fault Monitoring implementation of ping **MUST** be capable of transmitting Echo Request messages indefinitely.

[R19] An On-demand Fault Monitoring implementation of ping **MUST** allow the user to stop the transmission of Echo Request.

[R20] An On-demand Fault Monitoring implementation of traceroute **MUST** support the transmission of Echo Request messages to a unicast DA.

[R21] An On-demand Fault Monitoring implementation of traceroute **MUST** support the reception of Echo Reply messages from unicast addresses other than the target DA.

[R22] An On-demand Fault Monitoring implementation of traceroute **MUST** support reporting the IP addresses and TTL for each Echo Reply message received.

[R23] An On-demand Fault Monitoring implementation **MUST** allow the user to select the length of transmitted ICMP PDU.

[R24] An On-demand Fault Monitoring implementation of ping **MUST** support packet lengths of Echo Request message in the range of 64-1500 Bytes.

[D4] An On-demand Fault Monitoring implementation of ping **SHOULD** support packet lengths of Echo Request message in the range of 1501-10000 Bytes.

Recommended default settings are shown in Table 3.

On-Demand Tool		Recommended Default	Comments
ICMP Ping	Number of Echo Request Messages Transmitted	3	
	Echo Request Message Transmission Time Interval	1 second	
	Echo Request Message Length	64 Bytes	
ICMP Trac-	Echo Request Message	1 second	

eroute	Transmission Time Interval		
	Echo Request Message Length	64 Bytes	

528 **Table 3 - On-demand Tool Recommended Defaults**

529 SPs can use other on-demand tools such as TCP ping or HTTP ping in their networks. The use of these tools is outside the scope of the document.

531 ### 8.3 FM Reporting

532 The requirements for reporting of faults detected by Fault Monitoring for Proactive monitoring are described below.

534 **[R25]** FM implementations **MUST** support the ability to generate a notification to the SOF/ICM within 2 seconds of a fault being detected by an FM session.

536 **[R26]** A fault notification **MUST** contain the following attributes:

537 Date and Time of the fault

538 Source IP Address

539 Destination IP Address

540 FM Session ID if assigned by SOF

541 Notification Type

542 Notification Severity

543 Notification Description

544 [D5] An FM implementation **SHOULD** support synchronization of the local time-of-day clock with UTC to within one second of accuracy.

546 The Date and Time represent the Date and Time of the fault state change in UTC with millisecond granularity and comply with [D5] for accuracy.

548 The Source and Destination IP addresses are specified at the creation of the BFD session. These are transmitted in the measurement packets.

550 The FM Session ID can be assigned by the SOF upon the creation of the BFD session. This ID is not transmitted within any measurement packets and is used only by the SOF to identify an FM session.

The fault Notification Type is either SET or CLEAR. A SET is sent with all severities of notifications. A CLEAR is not sent with Informational Notifications.

The fault Notification Severity is either, Critical, Major, Minor, or Informational and is used to indicate the severity of the notification.

The fault Notification Description provides a textual description of the fault.

[R27] An FM implementation **MUST** support the ability to enable or disable notification of faults on a per FM session basis.

[R28] An FM implementation **MUST** support the ability to define the severity of a fault report.

[R29] An FM implementation **MUST** support at least two fault report severities, Critical and Major.

[O1] An FM implementation **MAY** support additional fault report severities.

The requirements for reporting of On-demand tools are described below.

[R30] A FM implementation of an ICMP Ping **MUST** report the following:

- Number of TX packets
- Number of RX packets
- Minimum Round Trip Delay
- Average Round Trip Delay
- Maximum Round Trip Delay
- Count of lost packets
- Percentage of lost packets

[R31] A FM implementation of an ICMP Traceroute **MUST** report the following for each response received to the ICMP Echo Request:

- IP Address
- Time to Live
- Round Trip Delay

9 Performance Management

Performance Management (PM) provides the ability to measure the performance of IP Services. This section contains the Use Cases, Tool Requirements, and Deployment Guidelines for PM for IP Services.

9.1 PM Use Cases

Degradations in performance can have a greater impact on customer's perception of network quality than faults. Most networks have failover mechanisms that provide protection in the event of a fault. In many cases, degradations do not cause these mechanisms to engage. As a result, customer packets may continue to be transported over degraded facilities, leading to retransmissions or excessive delay.

MEF 61.1 defines an IPVC Service Level Specification Attribute that allows objectives to be specified for a number of Performance Metrics such as One-way Mean Packet Delay and One-way Packet Loss Ratio. The performance objectives specified in the SLS are a commitment by the SP to the Subscriber of how the service is expected to perform and can result in SPs issuing rebates to Subscribers if SLS objectives are not met.

PM uses several terms that need to be understood.

- The first is SLS Reference Point (SLS-RP). This is defined in MEF 61.1 [33] as a point from or to which performance objectives are specified as part of an SLS; either an IPVC End Point or a location specified in the SLS Service Attribute.

- The second is Measurement Point (MP). An MP is defined within this document as a point from or to which performance is measured. An MP can be at an IPVC End Point or at a location specified by the SP. An MP is assigned an IP address and IP packets are routed between the IP addresses of two MPs. There are two types of MPs, Controller and Responder. A Controller MP is the MP that initiates SOAM PM Packets and receives responses from the Responder MP. A Responder MP is the MP that receives SOAM PM Packets from the Controller MP and transmits responses to the Controller MP. It should be noted that SLS-RP and MP of the same service and directionality, i.e., "from" or "to", may be co-located or placed in different points along the path of the service.

- The third term is an MP Pair. An MP Pair is a set of a particular Controller MP and a particular Responder MP that are measuring performance. An example is two MPs each located at different IPVC End Points of the same IPVC that are measuring performance between them. This MP Pair reports the performance between these two MPs as a part of the performance for the entire IPVC. An MP is a part of one or more MP Pairs.

- The fourth term is a PM Session. A PM Session is initiated on a Controller MP to take performance measurements for a given SOAM PM IP Traffic Class and a given Responder MP.

- The fifth term is Measurement Interval. Measurement Intervals (MI) are discrete, non-overlapping periods of time during which the PM Session measurements are performed and results are gathered.
- The sixth term is PM Tool. PM Tools are the functionalities or implementations that are used to perform the SOAM measurements. PM Tools are limited to TWAMP Light, STAMP, and TWAMP.
- Where the term PE is used in these figures this could represent a traditional PE, or a device or an application managed by the SP providing some or all of PE functionality.

Comment [MB1]: Make sure spelled out already

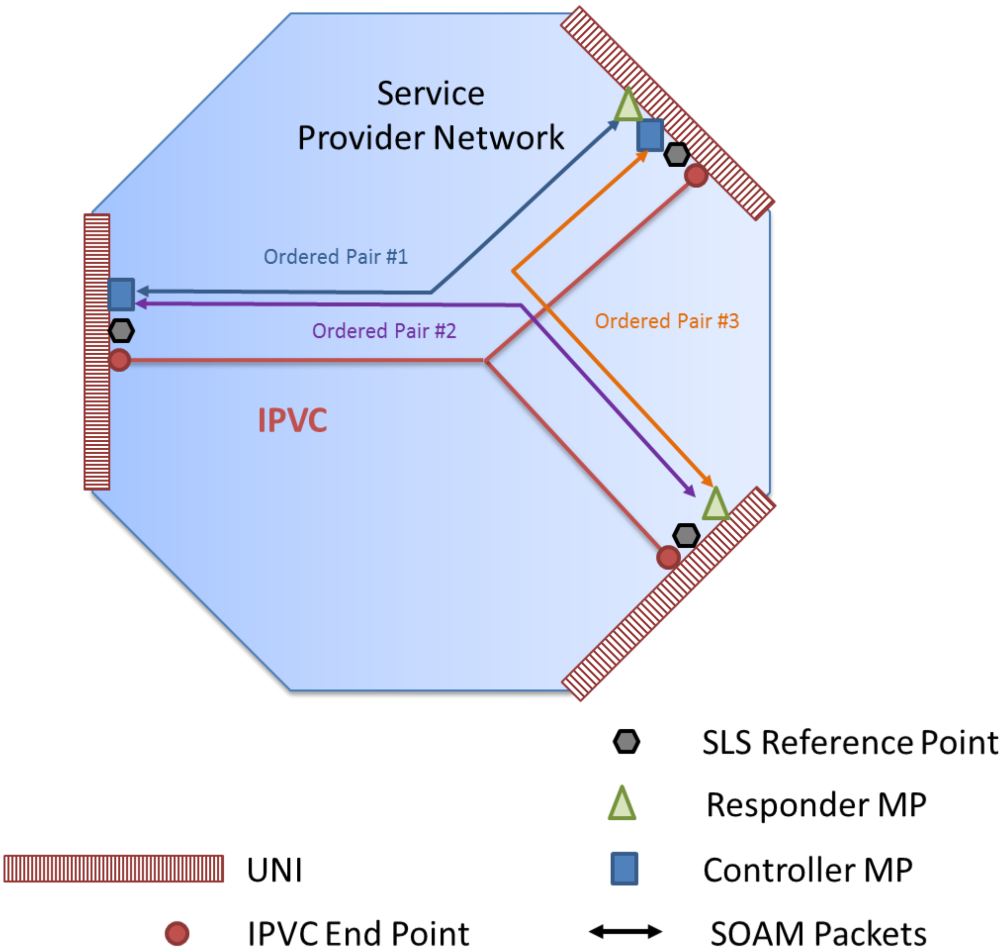

Figure 6 – SLS-RPs, MPs and Pair of MPs

Figure 6 shows a single IPVC. The SLS-RPs and MPs are located at the UNIs. Three Pairs of MPs are shown in blue, purple and orange. SOAM PM packets are exchanged between the MPs in each Pair of MPs.

SPs normally approach monitoring the performance of their services and network in one of two methods. In the first method, they identify IPVC End Points as SLS-RPs and configure MPs at each IPVC End Point including the entire path of the service in their SLS. In the second method, they designate SLS-RPs at some location, configure MPs at these locations, and measure performance between these MPs. Often with the second method there is an IPVC-like connection also known as an IP-PMVC (IP-Performance Monitoring Virtual Connection) dedicated to measuring the performance of connections between locations rather than monitoring specific Subscriber IPVCs. The difference between these is shown in Figure 7. Note that in both of these methods; MPs are created at the points in the network between which the SLS objectives are specified, i.e. in the same places as the SLS-RPs. This provides the most direct way of measuring performance so as to determine whether the objectives specified in the SLS have been met. However, it is not required that MPs and SLS-RPs are in the same places, and other arrangements are possible.

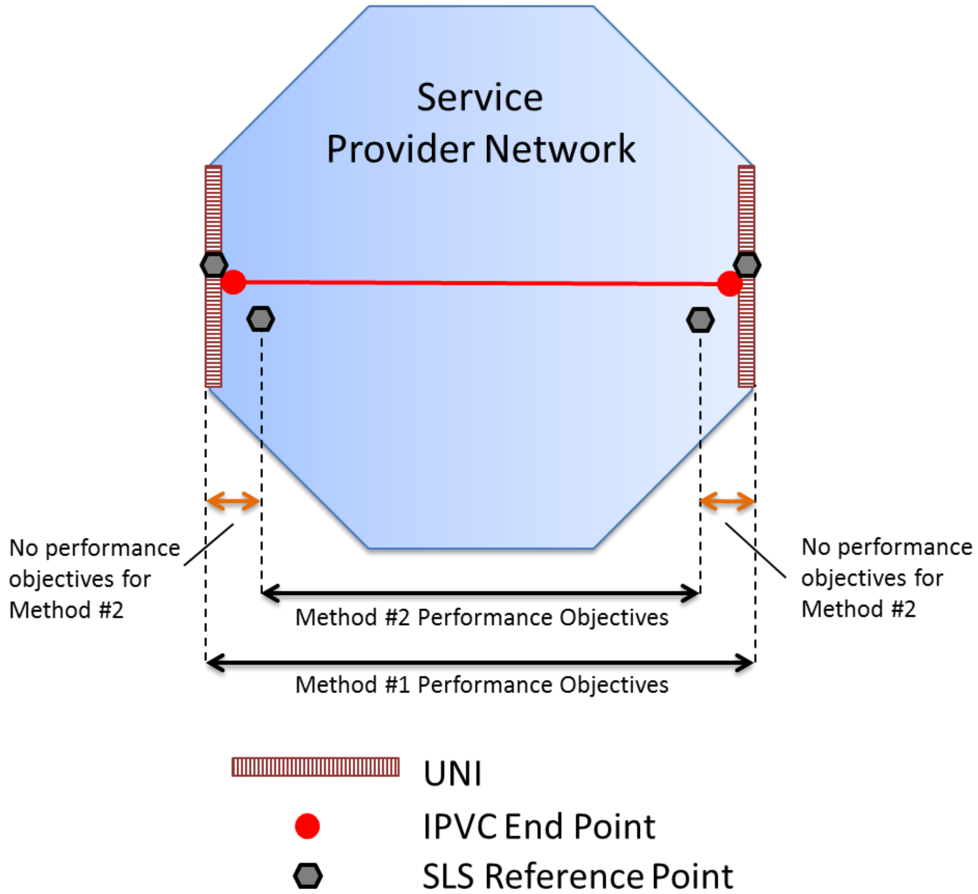

Figure 7 – SLS Method 1 and Method 2 Comparison

Examples of possible locations of the MPs are shown in Figure 8.

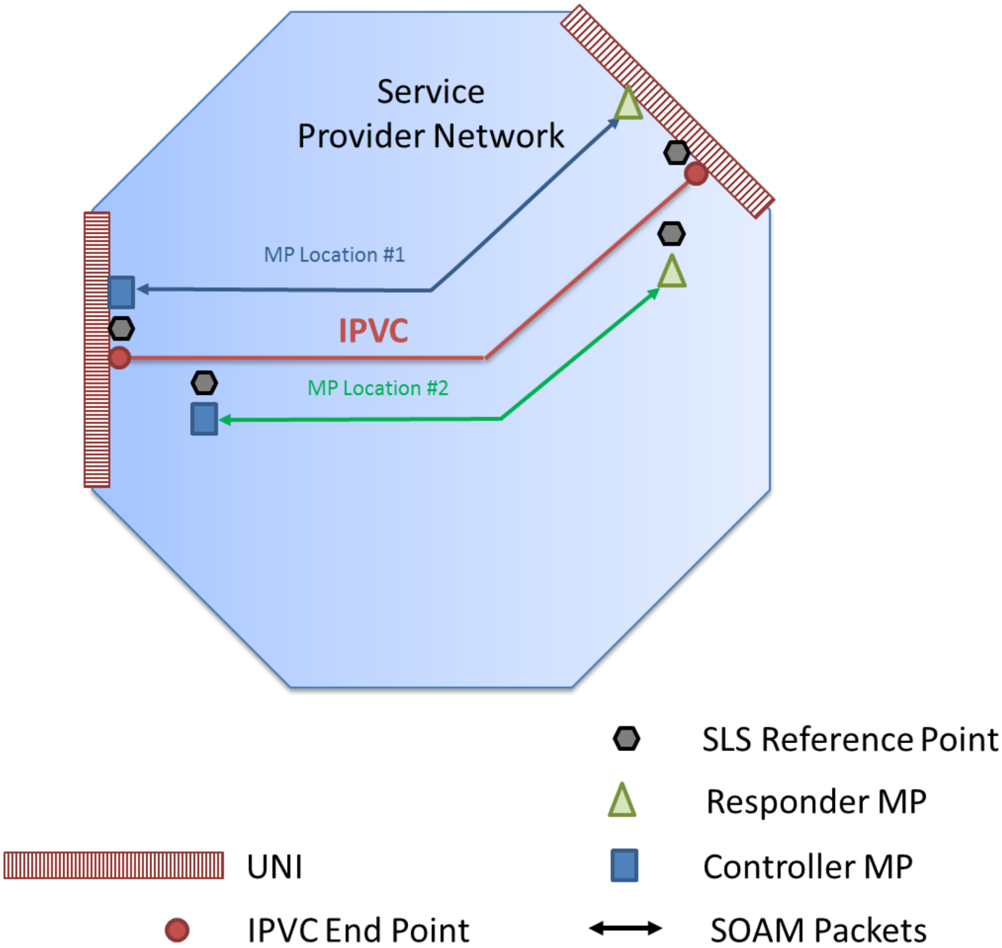

Figure 8 - Example MP Locations

PM can be performed using one of these three mechanisms:

- active method where synthetic packets are generated and measurements are performed on these packets
- passive method where counters reflecting customer traffic are retrieved from network elements
- hybrid method where customer traffic is modified to allow performance measurements to be performed using customer packets

This document focuses on active PM measurement and discusses hybrid PM measurement. Passive PM measurement is outside the scope of the document. This is because the retrieval of network element counters is implementation specific. Future versions of this document might address passive PM measurement if the retrieval of these counters is standardized.

Within this document, Active Measurement is specified as using TWAMP Light/STAMP/TWAMP. These PM tools are defined in RFC 5357 [10] and IETF Draft draft-ietf-ippm-stamp [20]. They enable Single-Ended monitoring of packet delay and packet loss. The protocol defined for each of these PM tools has a Session-Sender (Controller MP) and a Session-Reflector (Responder MP). The Controller MP generates measurement packets. The Responder MP responds to these packets. Time stamps in the packets allow one-way delay measurements to be performed if Time of Day (ToD) clock synchronization is present. If ToD synchronization is not present, it is not possible to make One-way delay measurements. Two-way delay measurements are possible and Two-way delay measurements can be divided in half as long as the results are identified as derived.

Hybrid Measurement is described using the AltM method. AltM is defined in RFC 8321 [17]. AltM enables Single-Ended monitoring for One-way Packet Delay and Packet Loss. See Section 10 for informational text on AltM.

PM Tools that measure Packet Delay (PD) and Packet Loss (PL) can be used to calculate additional metrics. PD measurements are used to calculate Mean Packet Delay, Inter-Packet Delay Variation, and Packet Delay Range. PL, measured as the difference between the number of transmitted packets and the number of packets received, is used to calculate the Packet Loss Ratio (PLR).

The following sections detail the use cases for PM including Location to Location monitoring and UNI to UNI monitoring. Location to Location monitoring provides a view of performance between locations using an IPVC-like connection but does not monitor any Subscriber IPVCs in a SP's network. UNI to UNI monitoring provides a view of the performance of a Subscriber IPVC from UNI to UNI.

9.1.1 Location to Location Monitoring

One way of monitoring performance by SPs is to monitor network performance from Location to Location via a single PE at each Location. As such, individual IPVCs are not monitored. Locations are connected together using a Network Measurement IPVC-like connection called an IP-Performance Monitoring Virtual Connection (IP-PMVC). This SLS monitoring via the Network Measurement IPVC-like connection between Locations provides an indication of the performance of the SPs network between the Locations. Authentication might be used to provide secure communications in TWAMP and STAMP implementations. If Active Measurement is being used the packets are routed over the Network Measurement IPVC-like connection that connects the Locations together. The measurement packets on the Network Measurement IPVC-like connection are expected to be treated similar to Subscriber packets. Service Providers need to ensure that they take into account network techniques such as Traffic Engineering (TE) and Equal Cost Multi Path (ECMP) routing when designing the operation of IP-PMVCs. Packet loss or delay that is measured between each location approximates the performance experienced by the Subscriber.

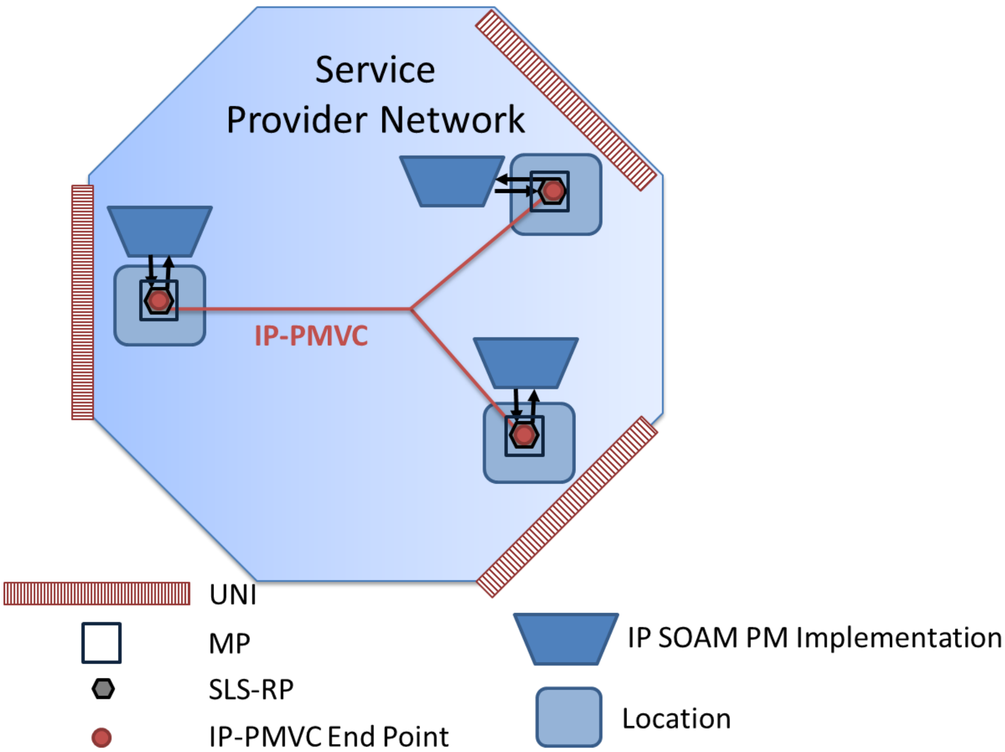

Figure 9 – Active PM Location to Location via IP-PMVC

Figure 9 is an example of a SP monitoring the performance of their network from Location to Location using an IP-PMVC dedicated to monitoring. The Locations are defined by the SP and interconnected using the IP-PMVC. An IP SOAM Implementation, either purpose built hardware, an application running in in a Virtual Machine (VM) on external hardware or an application running in the device at the location capable of generating measurement packets is connected to the SP network, sometimes via a UNI-like connection, and measurement packets are transmitted between all of the Locations via MPs that in this case are also IP-PMVC EPs. An MP can be the same point as the SLS-RP as shown in the figure but does not have to be the same point. Data collection is performed for some or all Pair of MPs.

An IP-PMVC is an IPVC-like connection between locations and is used for PM. The IP-PMVC can be routed similar to subscriber IPVCs. The IP-PMVC has EPs that are similar to an IPVC EP. A Location could represent a portion of a city, city, a country, a region or some other entity. A pair of MPs might include PM reports for multiple CoS Names that are monitored between the Locations. Subscribers who have IPVCs that connect between those entities might use the PM reports as an indication if the performance of their IPVCs has met the SLS. Within the SLS some Location Pairs might have different performance objectives than others. The SLS performance objectives that apply to one pair of MPs might be different than the SLS performance objectives that apply to another pair of MPs. This is because the expected performance between some cities, countries, or regions differs. Some Locations might offer higher performance SLS performance objectives while others offer lower performance SLS performance objectives.

In general, degradations that impact the Subscriber packets also impact the IP SOAM Performance monitoring packets.

9.1.2 IPVC Monitoring

Another method of PM for an IP service is to monitor the IPVC. This method might include the entire path of the service or some portion of it. Examples are from IPVC EP to IPVC EP or monitoring some portion of the IPVC. The SP is able to monitor degradations that occur at any point in the IPVCs between the two Measurement Points (MPs). This provides a more comprehensive view of the Subscriber's service performance. Using Active Measurement to perform IPVC monitoring requires that the PM packets be carried on the Subscriber's IPVC.

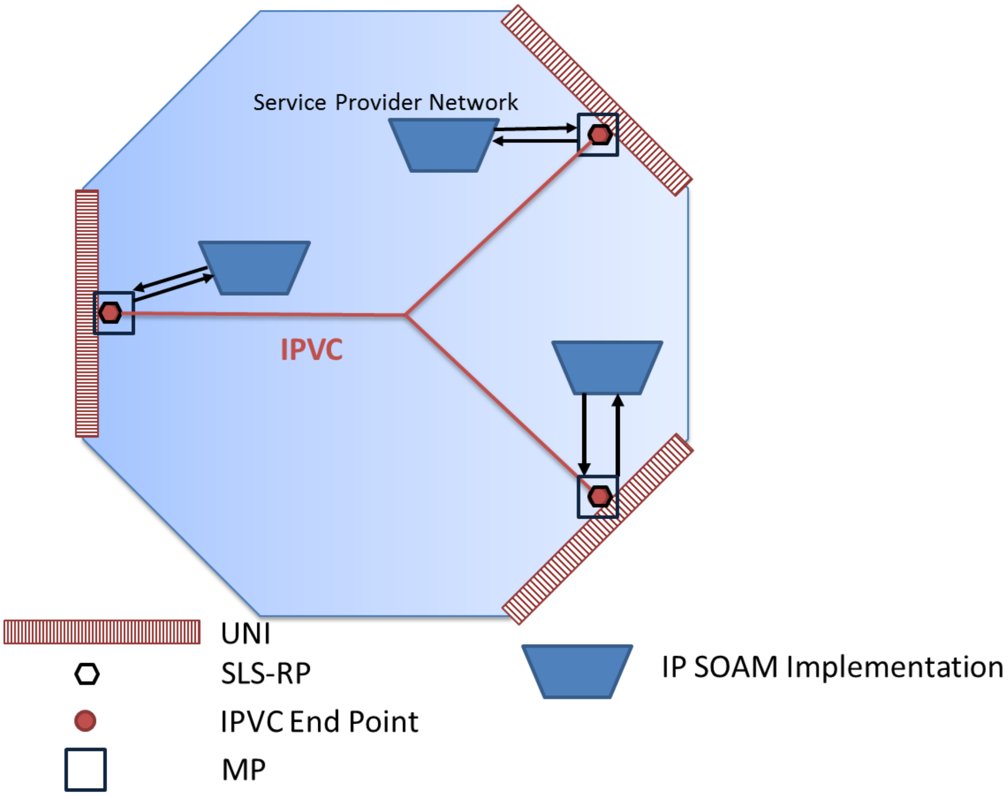

Figure 10 – IPVC EP to IPVC EP Active Measurement

Figure 10 is an example of Active Measurement on an IPVC from IPVC EP to IPVC EP. In this example, the IPVC EP, SLS-RP, and MP are all co-located. IP SOAM PM Implementations are deployed with the IPVC EPs. The IP SOAM PM Implementations are capable of generating monitoring packets. Packets are exchanged between all MPs active on the IPVC. Measurements between each Pair of MPs are made and collected.

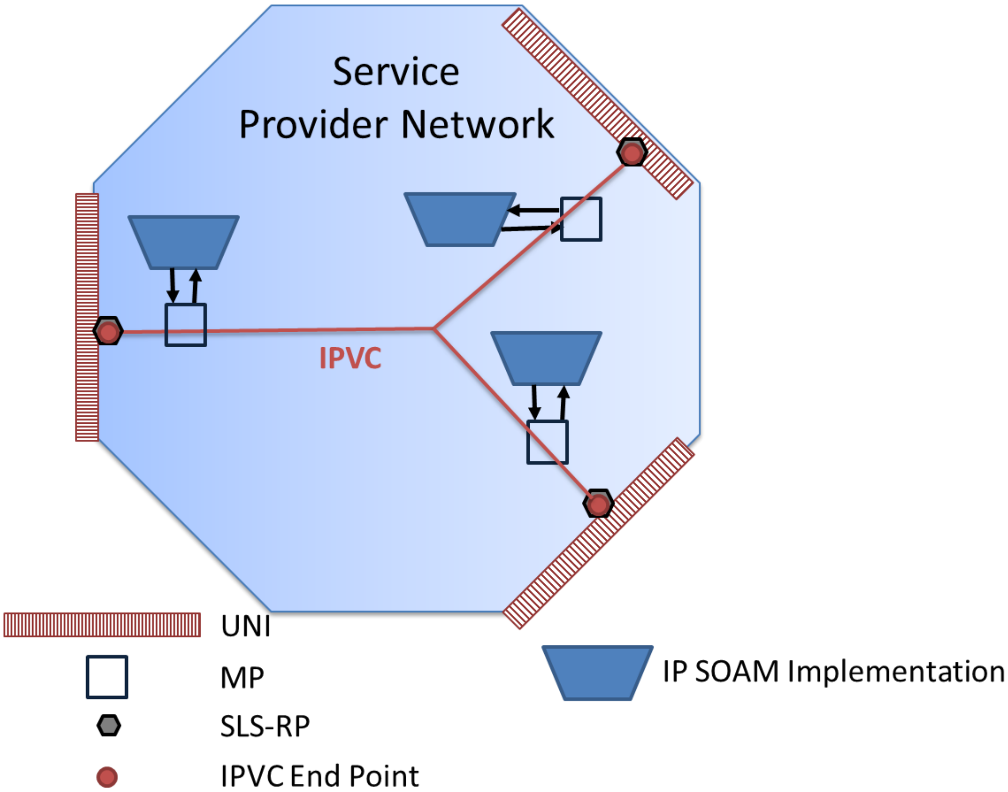

Figure 11 – Active Measurement when MPs are not at – IPVC EPs

Figure 11 shows monitoring of an IPVC that places the MPs at some point other than the IPVC EP. This is similar to Location to Location monitoring as shown in section 9.1.1 but monitoring is per Subscriber IPVC versus an IP-PMVC dedicated to monitoring. This type of monitoring requires support for MPs and IP SOAM Implementations at some point within the Service Provider's network.

While monitoring each IPVC has some definite benefits, it also has some challenges. IPVC monitoring requires that either that all IPVC EPs within an IPVC support both an MP and an IP SOAM PM Implementation, or that some points in the SP's network do so. This requires instantiation of many IP SOAM Implementations which can use processing capacity at each location.

This differs from Location to Location monitoring where only one or two IP-PMVC EPs per Location need to instantiate MPs and IP SOAM PM Implementations as shown in section 9.1.1. This limits the processing capacity required.

An IP SOAM PM Implementation might be able to be supported as a part of a device supporting the CE, PE, or other function rather than be a separate device as shown in the figures. Monitoring per IPVC EP increases the probe count compared to Location to Location monitoring and therefore increases the amount of data that must be processed.

A means to communicate between the ICM/ECM and the IP SOAM Implementation instantiated in the network is required. This can be accomplished via in-band or out-of-band methods. There are impacts of either of these communication methods. In-band communication could require additional bandwidth be provisioned to the device and out-of-band communication could require an additional service be configured to the device for communication. With Location to Location monitoring, this is limited to one or two probes versus bandwidth to every IPVC EP.

The functionality described above allows monitoring the performance between all IPVC EPs of an IPVC, between some subset of IPVC EPs, between IPVC EPs and MPs that are not at the IPVC EPs, and between any combination of these. These can be reflected as CE to PE, CE to CE, or PE to PE in more common terms.

9.2 PM Common Requirements

This section provides requirements that are applicable to PM. The requirements below provide for the Life Cycle (starting, stopping, etc.) and Storage.

Many requirements apply to an “IP SOAM PM Implementation”, which refers to the capabilities of a device or virtual function that are required to support IP SOAM Performance Monitoring.

9.2.1 Life Cycle

The requirements of this section apply to the life cycle of a PM Session, and to the scheduling of performance measurements conducted as part of a PM Session. Specifically, scheduling controls when, how long, and how often measurements will be taken for a PM Session.

9.2.1.1 General Overview of Parameters

The Performance Monitoring process is made up of a number of Performance Monitoring instances, known as PM Sessions. A PM Session is initiated on a Controller MP to take performance measurements for a given SOAM PM IP CoS Name and a given Responder MP. A PM Session is used for Loss Measurement and Delay Measurement.

The PM Session is specified by several direct and indirect parameters. A general description of these parameters is listed below, with more detailed requirements provided elsewhere in the document.

- The End Points are the Controller MP and a Responder MP.
- The DSCP used for the PM Session is chosen such that the performance of measurement packets is representative of the performance of the Qualified Packets being monitored.
- The PM Tool is any of the tools described in section 9.2 (TWAMP Light, STAMP, or TWAMP).

- The Message Period is the SOAM PM Packet transmission frequency (the time between SOAM PM Packet transmissions).
- The Start Time is the time that the PM Session begins.
- The Stop Time is the time that the PM Session ends.
- The Measurement Intervals are discrete, non-overlapping periods of time during which the PM Session measurements are performed and results are gathered. SOAM PM packets for a PM Session are transmitted only during a Measurement Interval. Key characteristics of Measurement Intervals are the alignment to the clock and the duration of the Measurement Interval. Measurement Intervals can be aligned to either the PM Session Start Time or to a clock, such as the local time-of-day clock. The duration of a Measurement Interval is the length of time spanned by a non-truncated Measurement Interval.
- The Repetition Time is the time between the start times of the Measurement Intervals.

9.2.1.2 Proactive and On-Demand PM Sessions

A PM Session can be classified as either a Proactive or an On-demand session. A Proactive session is intended to perpetually measure the performance between the MPs for the given SOAM PM IP CoS Name. An On-demand session is intended to monitor the performance for some finite period of time.

A Proactive session runs all the time once it has been created and started. Since the intent is to provide perpetual performance measurement, Proactive sessions use a Start Time of “immediate” and a Stop Time of “forever”. Measurements are collected into multiple fixed length Measurement Intervals covering different periods of time. Measurement Intervals for Proactive sessions are generally aligned to a clock, rather than the Session Start Time. Data is collected and a history of data is stored for a number of Measurement Intervals. Monitoring continues until the PM Session is deleted.

On-demand sessions are run when needed, and a report is provided at the end. Since On-demand sessions are intended to cover some finite period of time, absolute or relative Start and Stop Times may be used if those values are known. Alternatively, a Start Time of “immediate” and/or a Stop Time of “forever” may be used (with the intention of manually ending the session when no longer needed), especially if the monitoring period is of unknown duration (e.g., “until troubleshooting is completed”.) Measurements may be gathered into one Measurement Interval spanning the entire session duration, or multiple Measurement Intervals covering different periods of time. When multiple Measurement Intervals are used, then historical data from past Measurement Intervals may or may not be stored on the device. In addition, Measurement Intervals may be aligned with the session Start Time or aligned with a clock.

9.2.1.3 Create

A PM Session has to be created before it can be started. This applies for both On-demand and Proactive PM Sessions. In order to create a PM Session, a PM Tool must be assigned to the PM Session.

[D6] An IP SOAM PM Implementation **SHOULD** support multiple concurrent PM Sessions to the same destination, regardless of the setting of other parameters for the PM Sessions, and regardless of whether the PM Sessions use the same or different PM Tools using the five tuple (destination and source IP addresses, transport type, and destination and source port numbers) to identify each PM Session.

Multiple PM Sessions using the same PM Tool could be used, for example, to monitor different SOAM PM IP CoS Name (and hence measure performance for different IP CoS Name packets), different packet lengths, or to support both Proactive and On-demand sessions.

[R32] An IP SOAM PM Implementation **MUST** provide a way to indicate to the ICM/SOF whether a PM Session is Proactive or On-demand.

9.2.1.4 Delete

The requirements of this section apply to the deletion of a PM Session.

[R33] An IP SOAM PM Implementation **MUST** support the capability to delete a PM Session.

[R34] After a PM Session is deleted, further IP SOAM PM Packets relating to the session **MUST NOT** be sent.

[R35] After a PM Session is deleted, further measurements associated with the deleted PM Session **MUST NOT** be made.

[O2] Before the data from a deleted PM Session is lost, an IP SOAM PM Implementation **MAY** issue a report (similar to the report that would happen when Stop Time is reached).

[R36] After a PM Session is deleted, all the stored measurement data relating to the deleted PM Session **MUST** be deleted.

Note: a PM Session may be deleted at any point in its lifecycle, including before it has started.

9.2.1.5 Start and Stop

When a PM Session is started, it can be specified to start immediately, or be scheduled to start in the future. Both start conditions, particularly “immediate”, are conditional upon the local interface reaching the operational Up state and the address associated with the Responder being reachable.

[R37] For Proactive PM Sessions, the Start Time **MUST** be “immediate”.

[R38] For On-demand PM Sessions, an IP SOAM PM Implementation **MUST** support a configurable Start Time per PM Session. The Start Time can be specified as “immediate”, as an offset from the current time, or as a fixed absolute time in the future.

An offset from the current time (i.e., a "relative" time) could be specified as a given number of hours, minutes, and seconds from the current time. A fixed absolute time could be specified as a given UTC date and time.

[D7] For On-demand PM Sessions, the default Start Time **SHOULD** be “immediate”.

The following requirements apply to stopping of a PM Session.

[R39] For Proactive PM Sessions, the Stop Time **MUST** be “forever”.

[R40] For On-demand PM Sessions, an IP SOAM PM Implementation **MUST** support a configurable Stop Time per PM Session. The Stop Time can be specified as “forever” or as an offset from the Start Time.

An offset from the current time (i.e., a “relative” time) could be specified as a given number of hours, minutes, and seconds from the Start Time.

[R41] For On-demand PM Sessions, if the Stop Time is specified as an offset from the Start Time, then the Stop Time **MUST** be equal to or greater than the Message Period of the PM Session.

[D8] For On-demand PM Sessions, the default Stop Time **SHOULD** be "forever".

[R42] An IP SOAM PM Implementation **MUST** support stopping a PM Session by management action, prior to the Stop Time being reached.

[R43] After a PM Session is stopped, whether by reaching the scheduled Stop Time or by other means, further SOAM PM Packets relating to the session **MUST NOT** be sent.

[R44] After a PM Session is stopped, the stored measurements relating to the PM Session **MUST NOT** be deleted.

Note: a PM Session cannot be restarted once it has been stopped, as this would make it difficult to interpret the results. Instead, a new PM Session can be started.

9.2.1.6 Measurement Intervals

For the duration of a PM Session, measurements are partitioned into fixed-length Measurement Intervals. The length of the period of time associated with a Measurement Interval is called the duration of the Measurement Interval. The results of the measurements are captured in a Measurement Interval Data Set. The results in a Measurement Interval Data Set are stored separately from the results of measurements performed during other Measurement Intervals. This section contains requirements pertaining to Measurement Intervals in the Life Cycle of the PM Session. Requirements pertaining to storage of Measurement Interval Data Sets are found in section 9.2.2.1.

[R45] A SOAM PM Implementation **MUST** support a configurable duration for Measurement Intervals.

[R46] A SOAM PM Implementation **MUST** support a Measurement Interval with duration of 15 minutes for Proactive PM Sessions.

[R47] A SOAM PM Implementation **MUST** support Measurement Intervals with a duration of between 1 minute and 15 minutes (in 1 minute increments) for On-Demand PM Sessions.

[D9] The default Measurement Interval duration for On-Demand PM Sessions **SHOULD** be 5 minutes.

9.2.1.7 Repetition Time

For each PM Session, a Repetition Time can be specified if it is not desirable to perform measurements continuously. If the Repetition Time is “none”, then a new Measurement Interval is started immediately after the previous one finishes, and hence performance measurements are made continuously. If a Repetition Time is specified, a new Measurement Interval is not started until after Repetition Time has passed since the previous Measurement Interval started. During the time between the end of the previous Measurement Interval and the start of the next one, no SOAM PM Packets are sent by the Controller MP relating to the PM Session, and no measurements are initiated. Note that Responder MPs may send SOAM Packets during the time between two Measurement Intervals in response to SOAM Packets that may have previously been sent by the Controller MP.

[R48] An IP SOAM PM Implementation **MUST** support a configurable Repetition Time per PM Session. The Repetition Time can be specified as “none” or as a repeating time interval.

A repeating time interval (i.e., a relative time) could be specified as every given number of hours, minutes, and seconds from the Start Time.

[D10] The default Repetition Time **SHOULD** be “none”.

[R49] If the Repetition Time is a relative time, the time specified **MUST** be greater than the duration of the Measurement Interval.

[R50] During the time between two Measurement Intervals, SOAM PM Packets relating to the PM Session **MUST NOT** be sent by the Controller MP.

9.2.1.8 Alignment of Measurement Intervals

The following requirements pertain to the alignment of Measurement Intervals with time-of-day clock or PM Session Start Time.

[D11] An IP SOAM PM Implementation **SHOULD** by default align the start of each Measurement Interval, other than the first Measurement Interval, on a boundary of the local time-of-day clock that is divisible by the duration of the Measurement Interval (when Repetition Time is "none").

[D12] An IP SOAM PM Implementation **SHOULD** by default align the start of each Measurement Interval, other than the first Measurement Interval, on a boundary of the local time-of-day clock that is divisible by the Repetition Time (when Repetition Time is not "none").

When Measurement Intervals are aligned with the ToD clock, the Start Time of a PM Session might not correspond with the alignment boundary. In this case, the first Measurement Interval could be truncated.

[D13] An IP SOAM PM Implementation **SHOULD** allow for no alignment to the ToD clock.

[D14] An IP SOAM PM Implementation **SHOULD** support a configurable (in minutes) offset from ToD time for alignment of the start of Measurement Intervals other than the first Measurement Interval.

For example, if the Measurement Interval is 15 minutes and the Repetition Time is "none" and if ToD offset is 5 minutes, the Measurement Intervals would start at 5, 20, 35, 50 minutes past each hour.

9.2.1.9 Summary of Time Parameters

Possible values for the time parameters are summarized in the table below and are further explained in Appendix A:

Attribute	Possible Values	PM Session Type
Start Time	"Immediate" (default) ToD Offset Relative Time Fixed Time	Proactive or On-Demand Proactive or On-Demand On-Demand On-Demand
Stop Time	"Forever" (default) Relative Time	Proactive or On-Demand On-Demand
Repetition Time	"None" Relative Time	Proactive or On-Demand Proactive or On-Demand

Table 4 – Time Parameters

9.2.2 Storage

The requirements of this section apply to storage of performance measurement results taken during Measurement Intervals, using counters or Measurement Bins (for some delay-related parameters). Performance measurements are stored separately for each Measurement Interval. A Measurement Bin is a counter, and records the number of performance measurements falling within a specified range.

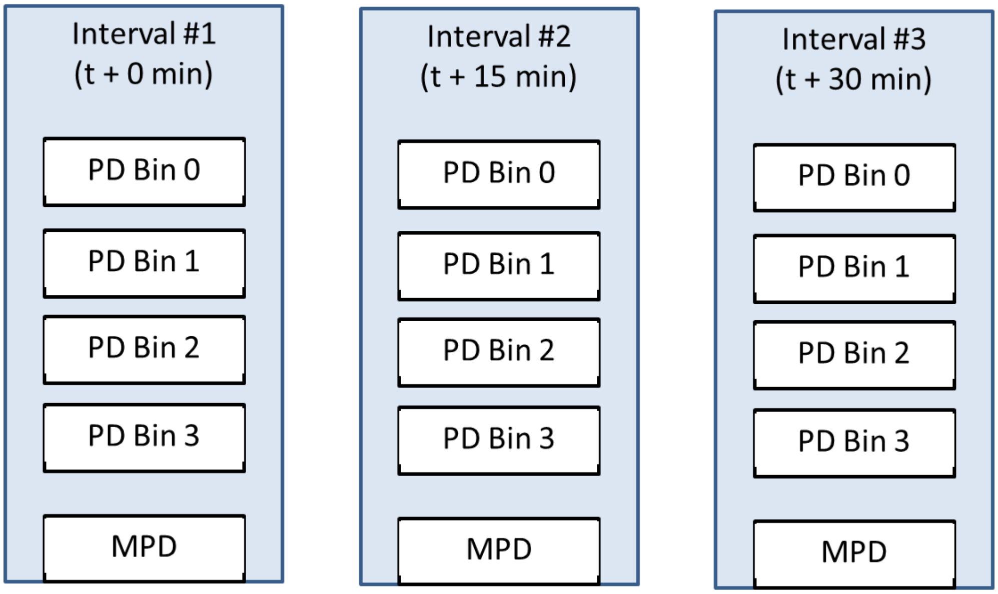

Figure 12 – Example of Measurement Bins and Intervals

Figure 12 shows the relationship between Measurement Bins and Measurement Intervals. Multiple Measurement Bins can be configured for a PM Session. Counts in these bins are incremented during each Measurement Interval.

Only delay measurements use bins; for loss measurements, bins are not used. Instead, each Measurement Interval contains counters that display Transmitted (TX) and Received (RX) packet counts. This is shown in Figure 13 below.

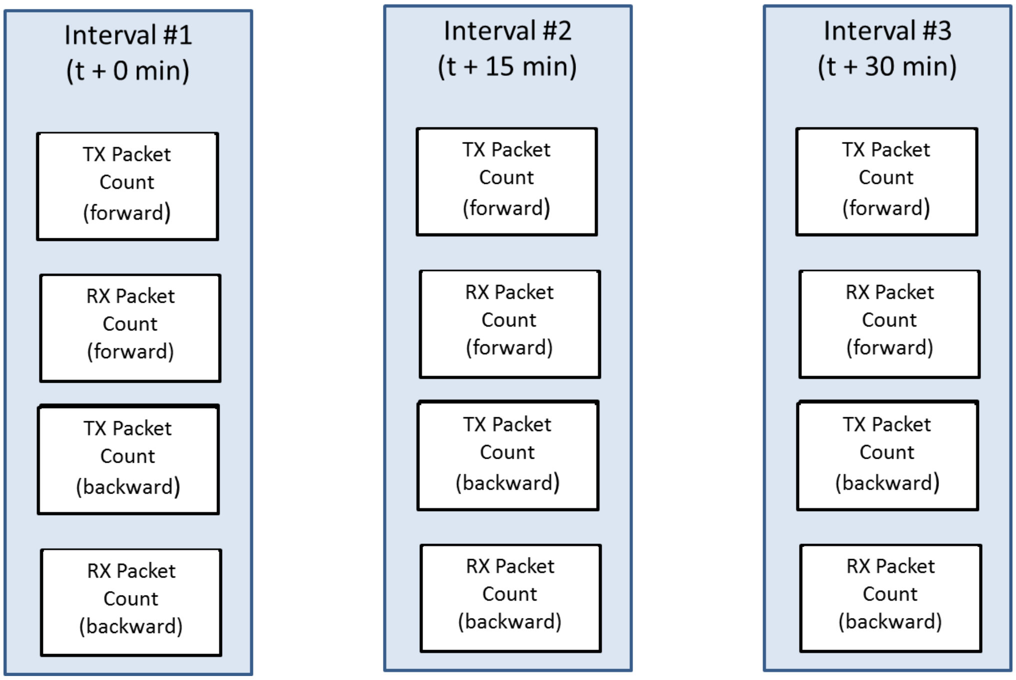

Figure 13 – Example of Packet Count Measurements

9.2.2.1 Measurement Interval Data Sets

The following requirements apply to the storage of the results of PD, PDR, MPD, IPDV, or PLR, performance measurements conducted between a given source and destination pair of MPs, for a given PM Session during a given Measurement Interval.

[R51] An IP SOAM PM Implementation **MUST** store measurement data for a current Measurement Interval and at least 8 hours of historic measurement data (captured per Measurement Interval) for a given data set of a Proactive PM Session.

[D15] An IP SOAM PM Implementation **SHOULD** store measurement data for a current Measurement Interval and at least 24 hours of historic measurement data (captured per Measurement Interval) for a given data set of a Proactive PM Session.

[D16] An IP SOAM PM Implementation **SHOULD** store measurement data for a current Measurement Interval and at least 8 hours of historic measurement data (captured per Measurement Interval) for a given data set of an On-demand PM Session.

[R52] An IP SOAM PM Implementation **MUST** record the value of the local ToD clock in UTC at the scheduled start of the Measurement Interval.

[R53] An IP SOAM PM Implementation **MUST** record the value of the local ToD clock in UTC at the scheduled end of the Measurement Interval.

[R54] An IP SOAM PM Implementation **MUST** support an elapsed time counter per Measurement Interval, which records the number of seconds that have elapsed since the Measurement Interval began.

[D17] An IP SOAM PM Implementation **SHOULD** support synchronization of the local time-of-day clock with UTC to within one second of accuracy.

[R55] An IP SOAM PM Implementation **MUST** record the results of a completed performance measurement as belonging to the Measurement Interval Data Set for the Measurement Interval in which the performance measurement was initiated.

[R56] An implementation of SOAM PM MUST support configurable wait timer, with the range of values from 1 second through to 5 seconds in one-second increments and the default value of 5 seconds, associated with the end of the Measurement Interval.

[R57] For Single-Ended Functions, a SOAM PM response packet received by the Controller MP after the expiration of the associated wait timer after the end of the Measurement Interval in which the corresponding SOAM PM request packet was transmitted **MUST** be discarded and considered lost.

9.2.2.2 Measurement Bins

The following requirements apply to the use of Measurement Bins for recording the results of delay performance measurements which can be used to determine conformance to PD, IPDV, and PDR objectives conducted between a given source and destination MP for a given PM Session during a Measurement Interval. Additional detail on Measurement Bins is provided in Appendix B.

The following requirements apply to each PD measurement supported in an IP SOAM PM Implementation.

[R58] An IP SOAM PM Implementation **MUST** support a configurable number of PD Measurement Bins per Measurement Interval.

[D18] For an IP SOAM PM Implementation, the default number of PD Measurement Bins per Measurement Interval **SHOULD** be 2.

[R59] An IP SOAM PM Implementation **MUST** support at least 2 PD Measurement Bins per Measurement Interval.

[D19] An IP SOAM PM Implementation **SHOULD** support at least 10 PD Measurement Bins per Measurement Interval.

The following requirements apply to each IPDV or PDR measurement supported in an IP SOAM PM Implementation.

[R60] An IP SOAM PM Implementation **MUST** support a configurable number of IPDV Measurement Bins per Measurement Interval.

[D20] For an IP SOAM PM Implementation, the default number of IPDV Measurement Bins per Measurement Interval supported **SHOULD** be 2.

[R61] An IP SOAM PM Implementation **MUST** support at least 2 IPDV Measurement Bins per Measurement Interval.

[D21] An IP SOAM PM Implementation **SHOULD** support at least 10 IPDV Measurement Bins per Measurement Interval.

[R62] An IP SOAM PM Implementation **MUST** support a configurable number of PDR Measurement Bins per Measurement Interval.

[D22] For an IP SOAM PM Implementation, the default number of PDR Measurement Bins per Measurement Interval supported **SHOULD** be 2.

[R63] An IP SOAM PM Implementation **MUST** support at least 2 PDR Measurement Bins per Measurement Interval.

[D23] An IP SOAM PM Implementation **SHOULD** support at least 10 PDR Measurement Bins per Measurement Interval.

Note: For PDR the minimum PD for the MI is subtracted before binning the results.

The following general Measurement Bin requirements apply to any IP SOAM PM Implementation. Each bin is associated with a specific range of observed delay, IPDV or PDR. Bins are defined to be contiguous, and each is configured with its lower bound. Because the bins are contiguous, it is only necessary to configure the lower bound of each bin. Furthermore, the lowest bin is assumed to always have a lower bound of 0, and the highest bin is assumed to have an upper bound of ∞.

Note: All values for IPDV, PDR and Two-way PD are positive by definition. Values for One-way PD can be negative if there is no ToD synchronization, and such measurements would not match any Measurement Bin as defined above; however, in this case taking One-way PD measurements is not recommended except for the purpose of finding the minimum PD for normalization of PDR, and finding the minimum PD does not require Measurement Bins.

A Measurement Bin is associated with a single counter that can take on non-negative integer values. The counter records the number of measurements whose value falls within the range represented by that bin.

[R64] An IP SOAM PM Implementation **MUST** support a configurable lower bound for all but the first Measurement Bin.

[R65] The lower bound for each Measurement Bin **MUST** be larger than the lower bound of the preceding Measurement Bin.

[R66] The unit for a lower bound **MUST** be in microseconds (μs).

[R67] The lower bound of the first Measurement Bin **MUST** be fixed to 0μs.

[R68] Measured performance values that are greater than or equal to the lower bound of a given bin and strictly less than the lower bound of the next bin (if any), **MUST** be counted in that, and only that bin.

[D24] The default lower bound for a Measurement Bin **SHOULD** be an increment of 5000 μs larger than the lower bound of the preceding Measurement Bin.

For example, four Measurement Bins gives the following:

Bin	Lower Bound	Range
Bin 0	0 μs	0 μs ≤ measurement < 5,000 μs
Bin 1	5,000 μs	5,000 μs ≤ measurement < 10,000 μs
Bin 2	10,000 μs	10,000 μs ≤ measurement < 15,000 μs
Bin 3	15,000 μs	15,000 μs ≤ measurement < ∞

Table 5 – Example Measurement Bin Configuration

[R69] Each Measurement Bin counter **MUST** be initialized to 0 at the start of the Measurement Interval.

9.2.2.3 Volatility

The following requirement applies to the volatility of storage for Measurement Interval data.

[D25] An IP SOAM PM Implementation **SHOULD** store the data for each completed Measurement Interval in local non-volatile memory.

The set of completed Measurement Intervals whose data is stored represents a contiguous and moving window over time, where the data from the oldest historical Measurement Interval is aged out at the completion of the current Measurement Interval.

9.2.2.4 Measurement Interval Status

The following requirements apply to a discontinuity within a Measurement Interval. Conditions for discontinuity include, but are not limited to, the following:

- Loss of connectivity between the Controller MP and the Responder MP.
- Per section 10.1.6.1 of ITU-T G.7710/Y.1701 [24], the local time-of-day clock is adjusted by at least 10 seconds.
- The conducting of performance measurements is started part way through a Measurement Interval (in the case that Measurement Intervals are not aligned with the Start Time of the PM Session).
- The conducting of performance measurements is stopped before the current Measurement Interval is completed.
- A local test, failure, or reconfiguration disrupts service on the IPVC.

[R70] An IP SOAM PM Implementation **MUST** support a Suspect Flag per Measurement Interval.

[R71] The Suspect Flag **MUST** be set to false at the start of the current Measurement Interval.

[R72] An IP SOAM PM Implementation **MUST** set the Suspect Flag to true when there is a discontinuity in the performance measurements conducted during the Measurement Interval.

Note: Loss of measurement packets does not affect whether the Suspect Flag is set.

[CD1]<[R72]When the suspect flag is set to true for a Measurement Interval, an IP SOAM PM Implementation **SHOULD** record the reason for the discontinuity.

[R73] The value of the Suspect Flag for a Measurement Interval **MUST** always be stored along with the other results for that Measurement Interval when that Measurement Interval's data is moved to history.

9.3 PM Implementation Requirements

A PM Implementation uses PM Tools to perform the measurements. A PM Session is an instantiation of a particular PM Tool within a PM Solution between a given pair of MPs using a given IP CoS Name over a given (possibly indefinite) period of time. A PM Session can be given a unique identifier, known as the PM Session ID, by the SOF. This is used by the SOF to identify a specific PM Session.

.

Note: Only unicast packets are used to perform PM Measurements to avoid causing congestion in the network.

An explanation of Single-Ended is shown in Figure 14. This term is also defined in MEF 35.1 [31].

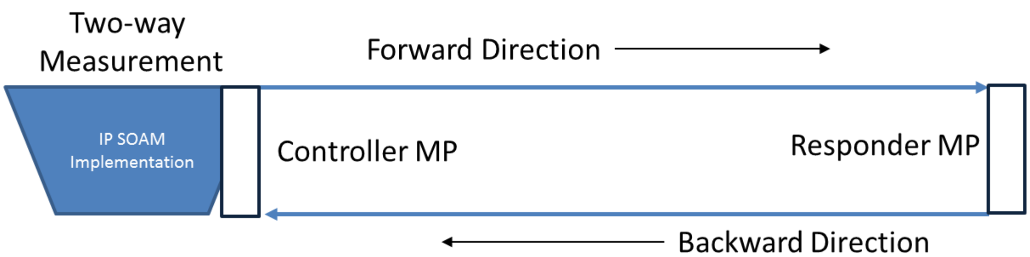

Figure 14 – Single-Ended Function

As seen in Figure 14, a Single-Ended Function places a Controller MP at one end of the service being monitored. The Controller MP transmits and receives measurement packets. The Single-Ended Function also places a Responder MP at the other end of the service being monitored. The Responder MP processes the packets received from the Controller MP and transmits packets to the Controller MP. Controller to Responder measurements and Responder to Controller measurements are also known as Forward and Backward measurements, respectively. Single-Ended Functions can be used to perform One-way measurement in the forward and backward directions, and to perform Two-way measurements. This is because the responder is not a simple loopback but processes the packets adding timestamps including the time the packet was received, the timestamp quality estimate, and the time the packet was transmitted as described in section 9.3.1. Single-ended forward and backward measurements are included in the scope of this document.

With optional time-of-day (ToD) clock synchronization, accurate One-way Packet Delay (PD) and Mean Packet Delay (MPD) measurements can be taken. Two-way PD, MPD, Packet Delay Range (PDR), and Inter-Packet Delay Variation (IPDV) measurements and One-way PDR and IPDV measurements can always be taken and do not require ToD clock synchronization. For PD and MPD, if ToD synchronization is not sufficiently accurate for performance measurement purposes, the One-way performance metrics of MEF 61.1 [33] can be estimated by dividing the Two-way measurement by 2, although this introduces considerable statistical bias. Also note that when measuring One-way PDR, it is necessary to normalize measurements by subtracting the minimum delay. This allows One-way PDR to be measured even if ToD synchronization is not present. Examples of this are shown below (more details in Appendix D).

When the minimum delay between two MPs is a positive value, use the lowest positive value as the minimum delay. For example, if the minimum delay measured between two MPs is 7000ms then all one-way delay measurements have 7000ms subtracted from them and the result is the normalized measurement.

When the minimum delay between two MPs is a negative value, use the most negative value as the minimum delay. For example, if the minimum delay measured between two MPs is -7000ms then all one-way measurements have -7000ms subtracted from them and the result is the normalized measurement.

MEF 61.1 [33] defines that multiple Class of Service Names (CoS Names) can be supported by an IP Service. These CoS Names are used to identify which CoS to map the packet to and how the packet is treated by the network. Each of the CoS Names can be used to specify a different objective within an SLS. When measuring the performance of an IP service, it might be necessary to monitor the performance of different CoS Names between the same two MPs. This is done by creating a separate PM Session for each CoS Name to be monitored. When the IP SOAM Measurement packets use the Subscriber IPVC they are treated the same way as the Subscriber packets for each CoS Name being monitored. When the IP SOAM Measurement packets use the IP-PMVC, they are treated the same as Subscriber packets for each CoS Name being monitored, though the IP-PMVC packets might travel on a different path than when PM is performed on the IPVC itself.

The intention is for IP SOAM Measurement packets to be treated the same as Subscriber IP Data packets and to take the same network paths. The IP SOAM Measurement packets include the DA of the IP SOAM Implementation at the targeted IPVC EP, CoS markings matching the Subscriber packets within the Service Provider's network for that CoS Name, and are introduced into the network onto the same device as the Subscriber's IP Data packets and that serves the Subscriber's IPVC EP. The IP SOAM Measurement packets use the same queues, processors, and network facilities as the Subscriber's IP Data packets. The IP SOAM Measurement packets experience the Service Provider's network in a similar manner to the Subscriber's IP Data packets.

In the case of Location to Location monitoring, the IP-PMVCs are configured similar to Subscriber IPVCs on devices serving Subscriber IPVCs. The SP needs to ensure IP SOAM Measurement packets are processed similarly to Subscriber IP Data packets. Using the same queues, processors, and network facilities as Subscriber packets can ensure that the IP SOAM Measurement packets experience the Service Provider's network in a similar manner to the Subscriber's.

Note: The Dual-Ended Function (OWAMP) is not within the scope of this document. OWAMP requires coordination and communication between the two ends of the service. Because of the added complexity of OWAMP vs TWAMP Light or STAMP, OWAMP is not addressed. One-way measurements are possible using a Single-Ended Function as discussed above.

9.3.1 PM Implementation Description

The PM Implementation provides Single-Ended Functions that measure Packet Delay (PD), and Packet Loss (PL). The implementation also provides calculations of Mean Packet Delay (MPD), Inter-Packet Delay Variation (IPDV), Packet Delay Range (PDR), and Packet Loss Ratio (PLR). The ability to use TWAMP Light to perform these measurements is mandatory, other tools can be used.

PD is measured using synthetic packets that are transmitted by the Controller MP with a Destination Address (DA) of the Responder MP with the time stamp (T1) set to the time the packet is transmitted. As described previously the Responder MP adds two time stamps (T2, T3) to the synthetic packets. The packets are transmitted by the Responder MP with the DA of the Controller MP. Upon receipt of the packets, the Controller MP adds an additional time stamp (T4) identifying the time the packet was received. Measurements and calculations using these time stamps are described in this section.

As noted above, the PD measurements are used to calculate several other metrics. The methodologies for these calculations are detailed below.

To determine the Mean Packet Delay the following formula is used:

$$\frac{\sum^{n}(Packet\ delays\ of\ all\ Packet\ Delay\ measurements\ in\ an\ MI)}{\sum^{n}(Total\ Packet\ Delay\ measurements\ of\ MI)}$$

Note: This is derived from MEF 35.1 [31].

To determine Inter Packet Delay Variation the following is used:

A parameter, *n*, is the IP SOAM Measurement packet ordered pair selection or offset as referred to in [D30]. Given a sequence of received periodic IP SOAM Measurement packets, the set of ordered pairs can be expressed as $\{\ \{p_1, p_{1+n}\}, \{p_2, p_{2+n}\}, \{p_3, p_{3+n}\}, \ ...\}$.

The IPDV is the calculated difference between each ordered pair selection.

IPDV is presented as a percentile for each MI. Various percentiles can be used. Recommendations are 95%, 99%, and 99.9%.

See Appendix D for a discussion of Packet Delay Range

PL is measured using the same synthetic packets transmitted to the same MPs (for more details see Appendix C). The number of packets transmitted by the Controller MP, the number of packets received at the Responder MP, the number of packets transmitted by the Responder MP, and the number of packets received by the Controller MP are collected. Calculations of One-way and Two-Way PLR are performed using these values. [R88] provides the formula used to calculate PLR based on the PL measurements.

Synthetic packets are inserted at a rate that provides statistically valid measurements. The synthetic packets have to be treated the same by the network as the Subscriber packets to obtain accurate results. In addition, the synthetic packets that are used for monitoring need to reflect the packet length of the CoS Name that is being monitored. As an example, a CoS Name that is intended for voice packets would use small packets while a CoS Name intended for file transfer might use longer packets.

[R74] An IP SOAM PM implementation **MUST** support TWAMP Light as a PM Tool.

[D26] An IP SOAM PM implementation **SHOULD** support STAMP as a PM Tool.

[O3] An IP SOAM PM implementation **MAY** support TWAMP as a PM Tool.

[CR1]<[R74] An implementation of a Controller MP in TWAMP Light mode MUST comply with all aspects of RFC 5357 [10], to the extent specified in Appendix I, that applies to the Session Sender.

[CR2]<[D26] An implementation of a Controller MP **MUST** comply with all aspects of IETF draft-ietf-ippm-stamp [20] that apply to the Session Sender when STAMP is used.

[CR3]< [O3] An implementation of a Controller MP **MUST** comply with all aspects of RFC 5357 [10] that apply to the Control Client and Session Sender, when TWAMP is used.

[CR4]<[R74]An implementation of a Responder MP in TWAMP Light mode **MUST** comply with all aspects of RFC 5357 [10], to the extent specified in Appendix I, that applies to the Session Reflector.

[CR5]<[D26] An implementation of a Responder MP **MUST** comply with all aspects of IETF draft-ietf-ippm-stamp [20] for a Session Reflector when STAMP is used.

[CR6]< [O3] An implementation of a Responder MP **MUST** comply with all aspects of RFC 5357 [10] for a Server and Session Reflector when TWAMP is used.

[R75] An IP SOAM PM Implementation **MUST** support a configurable transmission interval for measurement packets.

[R76] An implementation of a Controller MP **MUST** be able to transmit measurement packets at the following intervals: 100ms, 1second, 10seconds when TWAMP Light, STAMP, or TWAMP are being used.

[R77] An IP SOAM Implementation **MUST** support a mechanism to limit the number of IP SOAM PM packets processed per second.

[D27] An implementation of a Controller MP **SHOULD** be able to transmit measurement packets at the following interval: 10ms when TWAMP Light, STAMP, or TWAMP are being used.

[R78] An IP SOAM PM Implementation **MUST** support a configurable unicast destination IP address for measurement packets.

[R79] An IP SOAM PM Implementation **MUST** support the ability to set CoS Marking(s) for measurement packets.

[R80] An IP SOAM PM Implementation **MUST** support configurable IP packet length that includes the measurement PDU, further referred to as measurement packet lengths.

[R81] An IP SOAM PM Implementation **MUST** support measurement packet lengths in the range of 64-1500 Bytes.

[D28] An IP SOAM PM Implementation **SHOULD** support measurement packet lengths in the range of 1501-10000 Bytes.

[R82] When performing PM in IPv4 networks, the Do Not Fragment flag **MUST** be set to 1.

Avoiding fragmentation can be accomplished by ensuring that any generated packets are less than or equal to the MTU for the service.

[D29] An IP SOAM PM Implementation **SHOULD** support the configurable selection of pairs of measurement packets for IPDV measurement purposes.

[D30] The default selection offset for IPDV **SHOULD** be 1.

[R83] An IP SOAM PM Implementation **MUST** support the definition of one percentile for reporting IPDV at the end of each interval.

[D31] An IP SOAM PM Implementation **SHOULD** support the definition of three percentiles for reporting IPDV at the end of each interval.

[R84] An IP SOAM PM Implementation **MUST** support, for PDR measurement purposes, normalizing delays by subtracting the estimated minimum delay of the interval.

[D32] An IP SOAM PM Implementation **SHOULD** use the observed minimum delay of the previous Measurement Interval as the estimated minimum delay to normalize PDR measurements at the beginning of a Measurement Interval.

[D33] During the Measurement Interval an IP SOAM PM Implementation **SHOULD** set the estimated minimum to the lower of the previous estimate or the minimum measured delay for the current Measurement Interval.

A shift of the minimum delay might be significant, or it might be minor. The NE relies on the SOF/ICM to determine whether the change in the minimum is such that the PDR measurements for the Measurement Interval should be invalidated. In the case where the minimum has increased, the PDR measurements for the previous Measurement Interval may also need to be invalidated (see Appendix D for the detailed discussion).

TWAMP Light, STAMP, or TWAMP are used for Single-Ended PD and MPD measurements. Two-way delay measurements are performed by the Session-Sender using the timestamps in the Session-Reflector response packet. These timestamps are shown in Figure 15. Timestamp T1 is added by the Controller MP when the IP SOAM Measurement packet is transmitted. Timestamp T2 is added by the Responder MP when the IP SOAM Measurement packet is received. Timestamp T3 is added to the IP SOAM Measurement packet by the Responder MP when the packet is transmitted towards the Controller MP. Timestamp T4 is added to the IP SOAM Measurement packet by the Controller MP when the packet is received from the Responder MP.

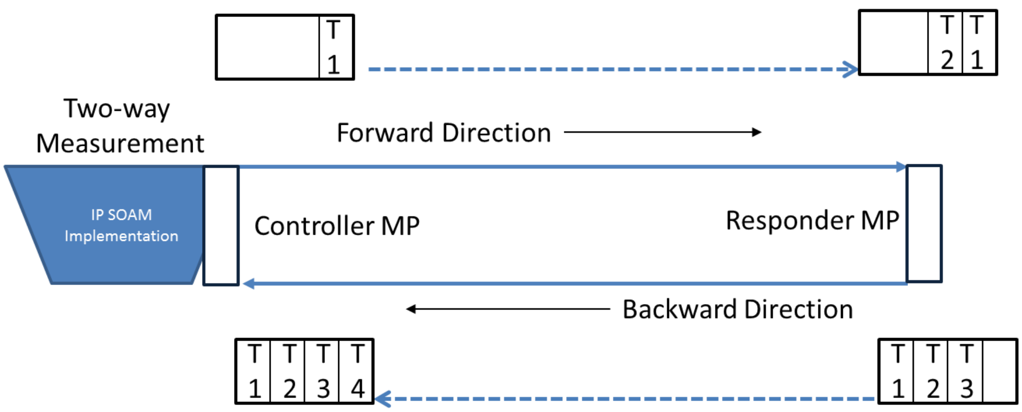

Figure 15 - Timestamp Locations

[R85] Two-way PD **MUST** be stated as (T4-T1)-(T3-T2) where T1 = Session-Sender Timestamp at the Controller MP, T2 = Receive Timestamp at the Reflector MP, T3 = Timestamp of packet transmit at the Reflector MP, and T4 = time measurement packet is received by Session-Sender (Controller MP) from Session-Reflector.

Note: By subtracting the difference between T3 and T2 the processing time at the Session-Reflector is removed from the measurement.

It is possible to measure One-way PD if ToD synchronization is in place between the MPs as described previously.

[R86] If ToD synchronization is in place, One-way PD **MUST** be stated as Forward PD (T2-T1) and Backward PD (T4-T3) where T1 = Session-Sender Timestamp at the Controller MP, T2 = Receive Timestamp at the Responder MP, T3 = Timestamp of packet transmit at the Responder MP, and T4 = time measurement packet is received by Session-Sender (Controller MP) from Session-Reflector.

[R87] If ToD synchronization does not exist between the MPs, one-way PD and MPD can be estimated by dividing the two-way measured value in half but the one-way value **MUST** indicate that this was the method used to obtain the value.

TWAMP Light, STAMP, or TWAMP are used to perform PL measurements. The PLR is the ratio of the number of packets lost to the number of packets transmitted by the Session-Sender.

[R88] The PLR **MUST** be determined using the following formula:

$$PLR = \frac{TX\ Packets - RX\ Packets}{TX\ Packets}$$

TWAMP Light, STAMP and TWAMP all support Stateful and Stateless Packet Loss measurements although the terms are only used in the STAMP working draft.

The definition of TWAMP Light as Stateful or Stateless is somewhat vague in RFC 5357 [10]. The TWAMP Light definition references section 4.2 of RFC 5357 [10] which defines the Session-Reflector as Stateful (e.g. adding timestamps and the sequence number to the response packet). For this reason this document specifies that TWAMP light is required to support Stateful Packet Loss measurement.

[R89] An IP SOAM PM Implementation using TWAMP Light **MUST** support Stateful Packet Loss measurement as specified in section 4.2 of RFC 5357 [10].

Stateful Packet Loss measurements require that the Session-Reflector (Responder MP) maintains test state determining forward loss, gaps recognized in the received sequence number. This implies that the Session-Reflector keeps a state for each PM session, uniquely identifying which SOAM PM Packets belong to one such PM session instance, and enabling adding a sequence number in the test reply that is individually incremented on a per-session basis. The method used by the Session-Reflector to keep a state for each PM Session is beyond the scope of this document.

Stateless Packet Loss measurements do not require the Session-Reflector (Responder MP) to maintain test state and Session-Reflector will reflect back the received sequence number without modification.

Stateful Packet Loss measurement allows One-way Packet Loss (Forward and Backward) to be measured. Stateless Packet Loss measurement allows only Two-way Packet Loss to be measured.

[R90] If an IP SOAM PM Implementation supports Stateful Packet Loss measurements, the Session-Controller (Controller MP) **MUST** identify the SOAM PM Packets belonging to each PM Session active at the Controller MP using the five tuples.

[R91] If an IP SOAM PM Implementation supports Stateful Packet Loss measurements, the Session-Reflector (Responder MP) **MUST** identify the SOAM PM Packets belonging to each PM Session active at the Responder MP using the five tuples.

[R92] An IP SOAM PM Implementation of STAMP **MUST** support Stateful Packet Loss measurements.

[R93] Two-way PLR **MUST** be calculated using the number of packets transmitted by the Session-Sender (Controller MP) and the number of packets received by the Session-Sender (Controller MP).

[R94] One-way PLR in the Forward direction **MUST** be calculated using the Sender Sequence Number of packets transmitted by the Controller MP, the Sequence Number of packets received by the Responder MP.

[R95] One-way PLR in the Backward direction **MUST** be calculated using the Sequence Number of the packets transmitted by the Responder MP and the total packets received at the Session-Sender (Controller MP).

The following requirements specify the ***output data set*** that is recorded by the Controller MP per Measurement Interval.

[R96] An IP SOAM PM implementation **MUST** provide the ability of the implementation to deliver PM reports to specified applications or user or the application or user to retrieve PM reports for each PM Session at the end of each PM Measurement Interval.

[R97] A PM report **MUST** contain the following in addition to the data shown in Table 6, Table 7, and Table 8:

- Controller IP Address
- Responder IP Address

The Controller and Responder IP Addresses might be changed to other identifiers within the LSO architecture.

[R98] The ability to retrieve all PM reports for a given PM Session **MUST** be provided.

[R99] A PM report **MUST** be available to be retrieved or delivered within two minutes of completion of the Measurement Interval x.

There may be packets in-flight between the Controller and Responder when the MI completes. This two minute period allows those packets to reach their destination and allows for processing of the PM data into the report format within the IP PM Implementation.

[R100] The ability to retrieve the current Measurement Interval **MUST** be provided. This displays the same information as the PM report up to the time of the query.

[R101] An IP SOAM PM Implementation **MUST** support the following data at the Controller MP per Measurement Interval per Stateful PM Session:

Data	Description
Start Time-of-day timestamp	A timestamp of the time-of-day in UTC at the scheduled start time of the Measurement Interval.
End Time-of-day timestamp	A timestamp of the time-of-day in UTC at the scheduled end time of the Measurement Interval.
Measurement Interval elapsed time	A counter of the number of seconds of the Measurement Interval as calculated by the NE. Note: this may differ from the difference between the start and end times if measurements started or stopped part way through the Measurement Interval, or if there was a shift in the time-of-day clock. Some of these conditions will result in the Suspect Flag being set.
Two-way PD counter per configured PD Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PD measurements that fall within the configured range.
Mean Two-way PD	A 32-bit integer reflecting the average (arithmetic mean) Two-way PD measurement in microseconds.
Minimum Two-way PD	A 32-bit integer reflecting the minimum Two-way PD measurement in microseconds.
Maximum Two-way PD	A 32-bit integer reflecting the maximum Two-way PD measurement in microseconds.
One-way IPDV counter in the Forward direction per configured IPDV Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of IPDV measurements (i.e., each instance of $\lvert D_i – D_j\rvert$ in the Forward direction) that fall within a configured bin.
Mean One-way IPDV in the Forward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way IPDV measurement in the Forward direction in microseconds.
Maximum One-way IPDV in the Forward direction	A 32-bit integer reflecting the maximum One-way IPDV measurement in the Forward direction in microseconds.
One-way IPDV counter in the Backward direction per configured IPDV Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of IPDV measurements in the Backward direction that fall within a configured bin.
Mean One-way IPDV in the Backward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way IPDV measurement in the Backward direction in microseconds.
Maximum One-way IPDV in the Backward direction	A 32-bit integer reflecting the maximum One-way IPDV measurement in the Backward direction in microseconds.
One-way PDR counter in the Forward direction per configured PDR Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PDR measurements in the Forward direction that fall within a configured bin.

Data	Description
Mean One-way PDR in the Forward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PDR measurement in the Forward direction in microseconds.
Maximum One-way PDR in the Forward direction	A 32-bit integer reflecting the maximum One-way PDR measurement in the Forward direction in microseconds.
One-way PDR counter in the Backward direction per configured PDR Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PDR measurements in the Backward direction that fall within a configured bin.
Mean One-way PDR in the Backward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PDR measurement in the Backward direction in microseconds.
Maximum One-way PDR in the Backward direction	A 32-bit integer reflecting the maximum One-way PDR measurement in the Backward direction in microseconds.
Minimum One-way PD in the Forward direction	A 32-bit integer reflecting the minimum One-way PD measurement in the Forward direction in microseconds.
Minimum One-way PD in the Backward direction	A 32-bit integer reflecting the minimum One-way PD measurement in the Backward direction in microseconds.
Tx Packet count in the Forward direction	A 32-bit counter reflecting the number of SOAM PM Packets transmitted in the Forward direction.
Rx Packet count in the Forward direction	A 32-bit counter reflecting the number of SOAM PM Packets received in the Forward direction.
Tx Packet count in the Backward direction	A 32-bit counter reflecting the number of SOAM PM Packets transmitted in the Backward direction.
Rx Packet count in the Backward direction	A 32-bit counter reflecting the number of SOAM PM Packets received in the Backward direction.

Table 6 – Mandatory Stateful Single-Ended Data Set

[R102] An IP SOAM PM Implementation **MUST** support the following data at the Controller MP per Measurement Interval per Stateless PM Session:

Data	Description
Start Time-of-day timestamp	A timestamp of the time-of-day in UTC at the scheduled start time of the Measurement Interval.
End Time-of-day timestamp	A timestamp of the time-of-day in UTC at the scheduled end time of the Measurement Interval.
Measurement Interval elapsed time	A counter of the number of seconds of the Measurement Interval as calculated by the NE. Note: this may differ from the difference between the start and end times if measurements started or stopped part way through the Measurement Inter-

Data	Description
	val, or if there was a shift in the time-of-day clock. Some of these conditions will result in the Suspect Flag being set.
Two-way PD counter per configured PD Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PD measurements that fall within the configured range.
Mean Two-way PD	A 32-bit integer reflecting the average (arithmetic mean) Two-way PD measurement in microseconds.
Minimum Two-way PD	A 32-bit integer reflecting the minimum Two-way PD measurement in microseconds.
Maximum Two-way PD	A 32-bit integer reflecting the maximum Two-way PD measurement in microseconds.
One-way IPDV counter in the Forward direction per configured IPDV Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of IPDV measurements (i.e., each instance of $\|D_i – D_j\|$ in the Forward direction) that fall within a configured bin.
Mean One-way IPDV in the Forward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way IPDV measurement in the Forward direction in microseconds.
Maximum One-way IPDV in the Forward direction	A 32-bit integer reflecting the maximum One-way IPDV measurement in the Forward direction in microseconds.
One-way IPDV counter in the Backward direction per configured IPDV Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of IPDV measurements in the Backward direction that fall within a configured bin.
Mean One-way IPDV in the Backward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way IPDV measurement in the Backward direction in microseconds.
Maximum One-way IPDV in the Backward direction	A 32-bit integer reflecting the maximum One-way IPDV measurement in the Backward direction in microseconds.
One-way PDR counter in the Forward direction per configured PDR Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PDR measurements in the Forward direction that fall within a configured bin.
Mean One-way PDR in the Forward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PDR measurement in the Forward direction in microseconds.
Maximum One-way PDR in the Forward direction	A 32-bit integer reflecting the maximum One-way PDR measurement in the Forward direction in microseconds.
One-way PDR counter in the Backward direction per configured PDR Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of PDR measurements in the Backward direction that fall within a configured bin.
Mean One-way PDR in the Backward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PDR measurement in the Backward direction in microseconds.
Maximum One-way PDR in the Backward direction	A 32-bit integer reflecting the maximum One-way PDR measurement in the Backward direction in

Data	Description
	microseconds.
Minimum One-way PD in the Forward direction	A 32-bit integer reflecting the minimum One-way PD measurement in the Forward direction in microseconds.
Minimum One-way PD in the Backward direction	A 32-bit integer reflecting the minimum One-way PD measurement in the Backward direction in microseconds.
Tx Packet count in the Forward direction	A 32-bit counter reflecting the number of SOAM PM Packets transmitted in the Forward direction.
Rx Packet count in the Backward direction	A 32-bit counter reflecting the number of SOAM PM Packets received in the Backward direction.

Table 7 – Mandatory Stateless Single-Ended Data Set

The minimum One-way PD measurements do not provide intrinsic information about the Packet Delay when time-of-day clock synchronization is not in effect, but are needed to detect changes in the minimum that may invalidate PDR measurements.

Note that when time-of-day clock synchronization is not in effect, measurements of One-way PD may result in a negative value for the minimum. This does not impact the ability to monitor changes in the minimum for the purpose of invalidating PDR measurements.

[R103] If time-of-day clock synchronization is in effect for both MPs in the Pair of MPs, an IP SOAM PM Implementation **MUST** be able to support the following additional data at the Controller MP per Measurement Interval per PM Session:

Data	Description
One-way PD counter in the Forward direction per configured PD Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of One-way PD measurements in the Forward direction that fall within the configured bin.
Mean One-way PD in the Forward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PD measurement in the Forward direction in microseconds.
Maximum One-way PD in the Forward direction	A 32-bit integer reflecting the maximum One-way PD measurement in the Forward direction in microseconds.
One-way PD counter in the Backward direction per configured PD Measurement Bin	A 32-bit counter per Measurement Bin that counts the number of One-way PD measurements in the Backward direction that fall within the configured bin.
Mean One-way PD in the Backward direction	A 32-bit integer reflecting the average (arithmetic mean) One-way PD measurement in the Backward direction in microseconds.

Table 8 – Mandatory Single-Ended Data Set with Clock Synchronization

9.4 PM Tool Requirements

The requirements for PM tools are detailed in this section. These requirements are currently limited to Active Measurement.

9.4.1 Active Measurement

Active Measurement uses synthetic packets to perform delay and loss measurements. Packets are generated by a Controller MP and are responded to by a Responder MP. Responder MPs are for Single-Ended Tools.

TWAMP Light/STAMP/TWAMP are the tools defined for Active Measurement. One-way Forward PD, One-way Backward PD, Two-way PD and Two-way packet counts can always be measured. From these measurements, Two-way MPD, One-way Forward IPDV, One-way Backward IPDV, One-way Forward PDR, One-way Backward PDR, and two-way PLR can always be calculated. If there is ToD synchronization between the Controller MP and the Responder MP, then One-way Forward MPD and One-way Backward MPD can also be calculated. If the Responder MP is stateful, then One-way Forward packet counts and One-Way Backward packet counts can be measured and from these measurements, One-way Forward PLR and One-Way Backward PLR can be calculated. If ToD synchronization is supported, One-way Forward PD, One-way Backward PD, One-way Forward MPD, and One-way Backward MPD, are supported.

The requirements for Active Measurement tools are defined in the following sections.

9.4.1.1 TWAMP Light

TWAMP Light is described in RFC 5357 [10] Appendix I. This is informative text in the RFC. Within the scope of this document, the support of TWAMP Light is required and therefore the text in the RFC is treated as if it was normative text. The method used as the Control-Client responder protocol is beyond the scope of this document.

TWAMP Light supports the same measurements as TWAMP but does not include the Control-Client that TWAMP requires. This makes TWAMP Light easier to implement and to deploy in a network. It does require that the two MPs in the Pair of MPs be configured so that the appropriate measurement packets are generated and collected. TWAMP Light test session may be performed in unauthenticated, authenticated or encrypted mode. In unauthenticated mode, no additional configuration is required. In Authenticated or encrypted mode, additional configuration of the Controller and Responder MPs is required to ensure that keys are correctly configured at both MPs. The TWAMP Light session is a stateful session. The method used for this configuration is beyond the scope of this document.

[R104] A TWAMP Light implementation **MUST** support a configurable UDP port number that the Controller MP transmits on and the Responder MP listens on.

[D34] A TWAMP Light implementation **SHOULD** support a default UDP port number that the Controller MP transmits on and the Responder MP listens on of 862.

9.4.1.2 STAMP

STAMP is an Active Measurement protocol for IP networks defined in draft-ietf-ippm-stamp [20]. It uses UDP encapsulation. Configuration and management of the STAMP Session-Sender, Session-Reflector and the test session between the two is outside the scope of this document.

STAMP test session may be performed in unauthenticated, authenticated or encrypted mode. In the unauthenticated mode STAMP is backward compatible with existing implementations of TWAMP Light (see more discussion on TWAMP Light in section 9.4.1.1).

A Stamp test session can detect packet re-ordering and duplication in the path between the STAMP Session-Sender and Session-Reflector. Measured performance metrics can be used to calculate additional performance metrics, e.g. percentile for forward packet delay or packet loss ratio.

9.4.1.2.1 Session-Sender Behavior

There are three modes of operation, Unauthenticated, Authenticated, and Encrypted, described for Session-Sender in draft-ietf-ippm-stamp [20].

[CR7]<[D26] A STAMP implementation **MUST** support the Session-Sender Unauthenticated Mode as specified in section 4.1.1 of draft-ietf-ippm-stamp [20].

[CD2]<[D26] A STAMP implementation **SHOULD** support the Session-Sender Authenticated Mode as specified in section 4.1.2 of draft-ietf-ippm-stamp [20].

[CR8]<[D26] A STAMP implementation **MUST** support a configurable UDP port that the Controller MP transmits on and the Responder MP listens on.

[CR9]<[D26] A STAMP implementation **MUST** support a default UDP port that the Controller MP transmits on and the Responder MP listens on of 862.

9.4.1.2.2 Session-Reflector Behavior

There are three modes of operation, Unauthenticated, Authenticated, and Encrypted, described for Session-Reflector in draft-ietf-ippm-stamp [20]. In addition, the Session-Reflector can be either Stateless (does not maintain test state) or Stateful (maintains test state). A Stateful Session-Reflector can be used to measure one-way packet loss. A Stateless Session-Reflector can be used to measure two-way packet loss only.

[CD3]<[D26] A STAMP implementation that supports Stateful mode **SHOULD NOT** support Stateless mode.

[CR10]<[D26] A STAMP implementation **MUST** support the Session-Reflector Unauthenticated Mode as specified in section 4.2.1 of draft-ietf-ippm-stamp [20].

[CD4]<[D26] A STAMP implementation **SHOULD** support the Session-Reflector Authenticated Mode as specified in section 4.2.2 of draft-ietf-ippm-stamp [20].

[CR11]<[D26] A STAMP implementation **MUST** support a configurable UDP port that the Responder MP listens on.

[CR12]<[D26] A STAMP implementation **MUST** support a default UDP port that the Responder MP listens on of 862.

9.4.1.2.3 Interoperability with TWAMP Light

In unauthenticated mode, a STAMP implementation can be interoperable with a TWAMP Light implementation. The Session-Reflector can support either TWAMP Light or STAMP and process packets correctly. The use of NTP timestamps by STAMP implementations make them interoperable with TWAMP Light implementations.

[CR13]<[D26] A STAMP implementation interoperating with TWAMP Light **MUST** use of NTP timestamps.

9.4.1.3 TWAMP

TWAMP is defined in RFC 5357 [10]. TWAMP includes a control protocol and a test packet definition. The TCP control protocol allows for the configuration of a test between a Session-Sender and a Session-Reflector. It defines a Control Server and a Control Client. The test packet defines the packets exchanged between the Session-Sender and the Session-Reflector.

[CR14]<[O3] A TWAMP implementation **MUST** comply with security recommendations in section 6 of RFC 5357 [10].

9.4.1.3.1 Session-Sender Behavior

There are three modes of operation, Unauthenticated, Authenticated, and Encrypted, described for Session-Sender in RFC 5357 [10].

[CR15]<[O3] A TWAMP implementation **MUST** support the Session-Sender Unauthenticated Mode as specified in section 4 of RFC 5357 [10].

[CD5]<[O3] A TWAMP implementation **SHOULD** support the Session-Sender Authenticated Mode as specified in section 4 of RFC 5357 [10].

[CD6]<[O3] A TWAMP implementation SHOULD support the Session-Sender Encrypted Mode as specified in section 4 of RFC 5357 [10].

[CR16]<[O3] A TWAMP implementation **MUST** support a configurable UDP port that the Controller MP transmits on.

[CR17]<[O3] A STAMP implementation **MUST** support a default UDP port that the Controller MP transmits on.

9.4.1.3.2 Session-Reflector Behavior

There are three modes of operation, Unauthenticated, Authenticated, and Encrypted, described for Session-Reflector in RFC 5357 [10].

[CR18]<[O3] A TWAMP implementation **MUST** support the Session-Reflector Unauthenticated Mode as specified in section 4 of RFC 5357 [10].

[CD7]<[O3] A TWAMP implementation **SHOULD** support the Session-Reflector Authenticated Mode as specified in section 4 of RFC 5357 [10] .

[CD8]<[O3] A TWAMP implementation SHOULD support the Session-Reflector Encrypted Mode as specified in section 4 of RFC 5357 [10].

[CR19]<[O3] A TWAMP implementation **MUST** support a configurable UDP port that the Responder MP listens on.

[CR20]<[O3] A STAMP implementation **MUST** support a default UDP port that the Responder MP listens on of 862.

9.5 Threshold Crossing Alerts (TCAs)

Performance thresholds, and corresponding Threshold Crossing Alerts (TCAs), can be configured for certain performance metrics, and used to detect when service performance is degraded beyond a given pre-configured level. Thresholds are always specific to a particular performance metric and a particular PM Session. When the measured performance in a Measurement Interval for that session reaches or exceeds the configured threshold level, a TCA can be generated and sent to an ICM or SOF.

In normal operation, performance data is collected from a device or network function by the ICM/SOF either periodically (e.g. once an hour) or On-demand. TCAs can be used as warning notifications to the ICM/SOF of possible service degradation, thus allowing more timely action to further investigate or address the problem. For example, if the maximum One-way PD threshold was set to 10ms, and a One-way PD value was measured at more than 10ms, a TCA would be generated.

[O4] An IP SOAM PM Implementation **MAY** support Threshold Crossing Alert functionality as described in sections 9.5.1, 9.5.2, and 9.5.3.

[O5] An IP SOAM PM Implementation **MAY** allow the time period for a TCA to be defined differently than the MI of the associated PM Session. As an example a TCA of five minutes could be defined even though there is a MI of 15 minutes for a particular PM Session.

The requirements in the following subsections only apply if TCA functionality is supported.

9.5.1 TCA Reporting

Thresholds and associated TCAs are specific to a particular performance metric in a given PM Session. There are two types of TCA reporting: stateless and stateful. With stateless reporting, a TCA is generated in each Measurement Interval in which the threshold is crossed. With stateful reporting, a SET TCA is generated in the first Measurement Interval in which the threshold is crossed, and a CLEAR TCA is subsequently generated at the end of the first Measurement Interval in which the threshold is not crossed.

Note: In ITU-T G.7710 [24] terminology, stateless TCA reporting corresponds to a transient condition, and stateful TCA reporting corresponds to a standing condition.

Regardless of the type of TCA reporting (stateless or stateful), it is not desirable to generate more than one TCA for a given threshold during each Measurement Interval, as to do otherwise could cause unnecessary load both on the NE and on the ICM/SOF receiving the TCAs.

Thresholds and TCAs are only defined for certain performance metrics, as described in section 9.5.2. Note that all of these performance metrics have the property that the value cannot decrease during a given Measurement Interval.

The process that takes a given threshold configuration for a given performance metric in a given PM Session and generates corresponding TCAs is termed a TCA Function. Multiple TCA Functions with different threshold values can be configured for the same PM Session and performance metric, so that TCAs can be generated for different degrees of service degradation. Where multiple TCA Functions are configured, corresponding TCAs are generated independently for each TCA Function.

9.5.1.1 Stateless TCA Reporting

The stateless TCA reporting treats each Measurement Interval separately. When using stateless TCA reporting, each TCA Function has a single configured threshold. As soon as the threshold is reached or crossed in a Measurement Interval for a given performance metric, a TCA is generated.

The following figure illustrates the behavior of stateless TCA reporting.

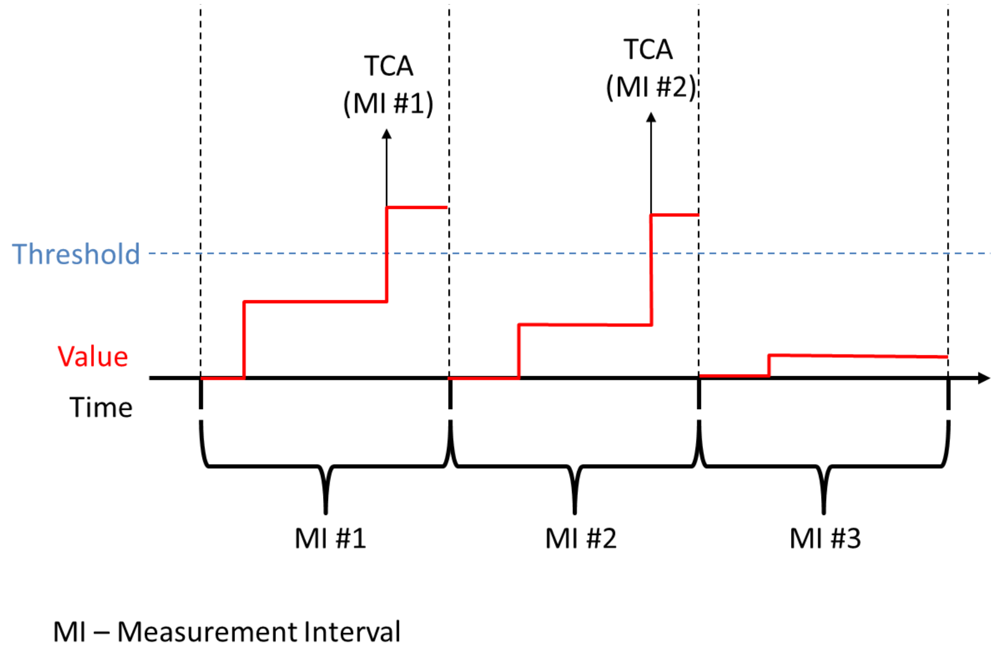

Figure 16 – Stateless TCA Reporting Example

As shown in the example in Figure 16, in MI #1, the measured performance value (e.g., Maximum Packet Delay) crosses the corresponding threshold. Therefore a TCA is generated for MI #1. In MI #2, this threshold is crossed again. Another TCA is generated for MI #2. In MI #3, the measured performance value doesn't reach the threshold. There is no TCA for that performance metric for MI #3.

9.5.1.2 Stateful TCA Reporting

Stateful TCA reporting is another option for how TCAs are generated, that can reduce the total number of TCAs. The intent is to provide a notification when a degradation is first encountered, followed by another when the problem is resolved. This contrasts with stateless TCA reporting, in which TCAs are generated continuously for as long as the degradation lasts.

When using stateful TCA reporting, each TCA Function has two configured thresholds: a SET threshold and a CLEAR threshold. These may be the same, or the CLEAR threshold may be lower than the SET threshold. The TCA Function also has an internal state, which may be 'set' or 'clear'.

The TCA Function begins in the 'clear' state. A SET TCA is generated in the first Measurement Interval as soon as the SET threshold is reached or exceeded. The TCA Function is then considered to be in a 'set' state, and no further SET TCAs are generated in this state. In each subsequent Measurement Interval in which the CLEAR threshold is reached or exceeded, no TCA is generated.

At the end of the first Measurement Interval in which the CLEAR threshold is not reached or exceeded, a CLEAR TCA is generated, and the TCA Function returns to the 'clear' state. Thus, each SET TCA is followed by a single CLEAR TCA.

The following figure shows an example of stateful TCA reporting. In this example, the CLEAR threshold is equal to the SET threshold.

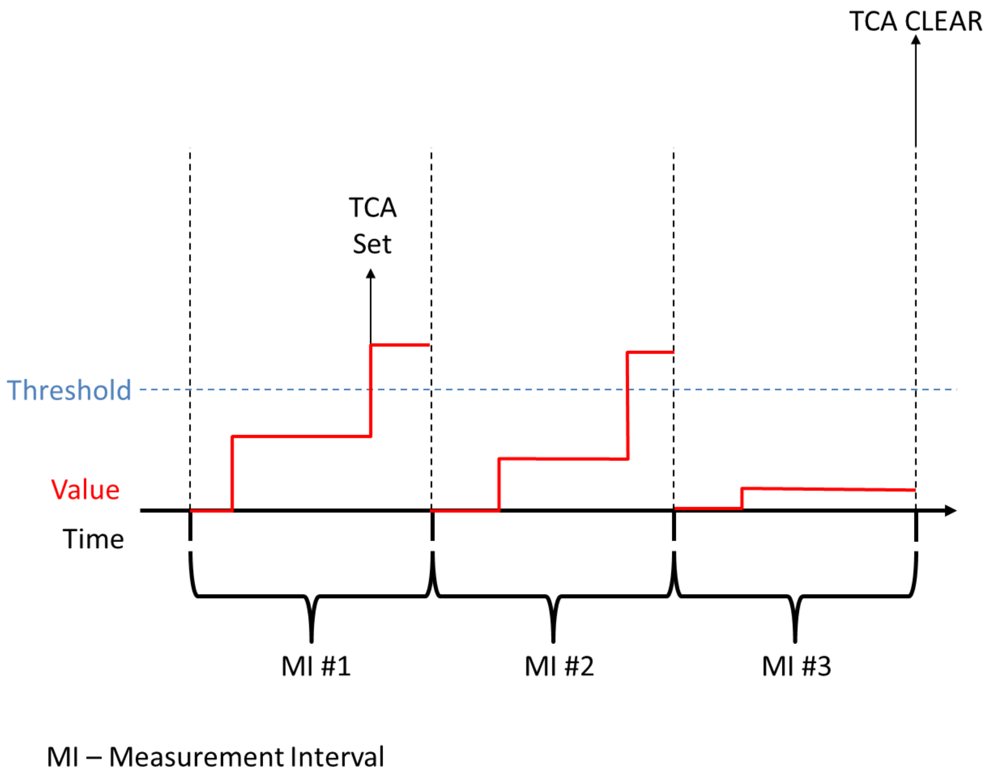

Figure 17 – Stateful TCA Reporting Example

In the example in Figure 17, a SET TCA is generated in MI #1. In MI #2, the threshold is crossed again but no SET TCA is generated because a SET TCA had been generated in MI #1. MI #3 is the first subsequent Measurement Interval that the measured performance value is below the CLEAR threshold. A CLEAR TCA is generated at the end of MI #3.

9.5.2 SOAM PM Thresholds for TCAs

TCAs are useful for some performance metrics but may not be meaningful for others. This section describes which performance metrics are required and how to support TCAs.

For performance metrics that use Measurement Bins, thresholds are defined in terms of an Upper Bin Count (UBC). The Upper Bin Count of bin k is the total of the counts for bins k and above, i.e. UBC(k) = count of bin (k) + count of bin (k+1) + ... + count of bin (n), where n is the last bin.

To configure a threshold, both the bin number, k, and the total count, N, need to be specified – this is represented as (N, k). A threshold (N, k) is considered to have been crossed when UBC(k) >= N. Figure 18 illustrates how a threshold is configured using bins.

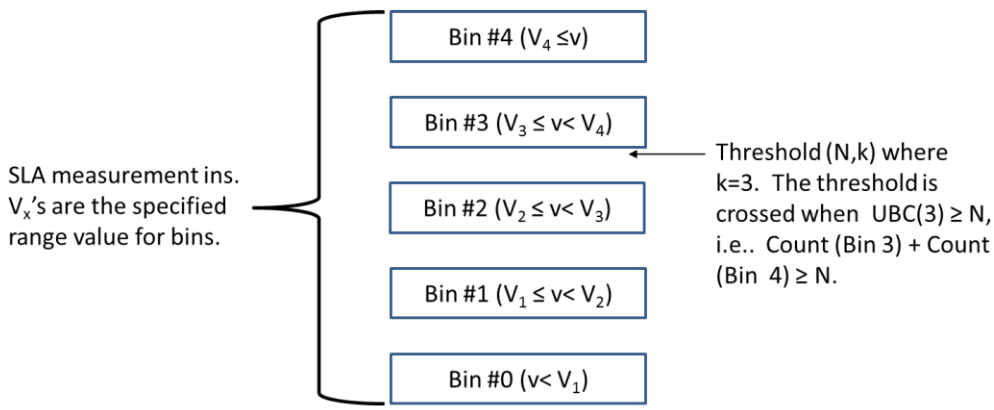

Figure 18 – Upper Bin Count for Threshold Crossing

The following table lists the applicable performance metrics that support TCAs. In each case, both One-way, and where applicable, Two-way performance metrics can be used. The table describes in each case the parameters that must be configured for the threshold, and the definition of when the threshold is crossed. For stateful TCA reporting, the "SET" thresholds and "CLEAR" thresholds are defined in the same way (although the configured values may be different).

Performance Metric	**Configured Threshold**	**Threshold Crossing Detection**	**Notes**
One-way IPDV in the Forward direction	Forward One-way (N_{IPDV}, k)	UBC(k) ≥ Forward one-way N_{IPDV}	Using Measurement Bins
One-way Maximum IPDV in the Forward direction	Forward One-way ($V_{maxIPDV}$)	Max IPDV ≥ Forward One-way $V_{maxIPDV}$	
One-way IPDV in the	Backward One-way	UBC(k) ≥ Backward	Using Measurement

Performance Metric	**Configured Threshold**	**Threshold Crossing Detection**	**Notes**
Backward direction	(N_{IPDV}, k)	one-way N_{IPDV}	Bins
One-way Maximum IPDV in the Backward direction	Backward One-way ($V_{maxIPDV}$)	Max IPDV ≥ Backward One-way $V_{maxIPDV}$	
One-way PD in the Forward direction	Forward One-way (N_{PD}, k)	UBC(k) ≥ Forward one-way N_{PD}	Using Measurement Bins. Requires ToD Synchronization
One-way Maximum PD in the Forward direction	Forward One-way (V_{maxPD})	Max PD ≥ Forward One-way V_{maxPD}	Requires ToD Synchronization
One-way PD in the Backward direction	Backward One-way (N_{PD}, k)	UBC(k) ≥ Backward one-way N_{PD}	Using Measurement Bins. Requires ToD Synchronization
One-way Maximum PD in the Backward direction	Backward One-way (V_{maxPD})	Max PD ≥ Backward One-way V_{maxPD}	Requires ToD Synchronization
Two-way PD	Two-way (N_{PD}, k)	UBC(k) ≥ Two-way N_{PD}	Using Measurement Bins
Two-way Maximum PD	Two-way V_{maxPD}	Max PD ≥ Two-way V_{maxPD}	
One-way PDR in the Forward direction	Forward One-way (N_{PDR}, k)	UBC(k) ≥ Forward one-way N_{PDR}	Using Measurement Bins
One-way Maximum PDR in the Forward direction	Forward One-way (V_{maxPDR})	Max PDR ≥ Forward One-way V_{maxPDR}	
One-way PDR in the Backward direction	Backward One-way (N_{PDR}, k)	UBC(k) ≥ Backward one-way N_{PDR}	Using Measurement Bins
One-way Maximum PDR in the Backward direction	Backward One-way (V_{maxPDR})	Max PDR ≥ Backward One-way V_{maxPDR}	

Performance Metric	Configured Threshold	Threshold Crossing Detection	Notes
One-way Lost Packets (LP) in the Forward direction	Forward One-way (N_{LP})	LP ≥ Forward one-way N_{LP}	The count of Lost Packets is determined the following formula: TX packet count Forward direction – RX packet count Forward direction = Lost Packet count Forward direction
One-way Lost Packets (LP) in the Backward direction	Backward One-way (N_{LP})	LP ≥ Backward one-way N_{LP}	The count of Lost Packets is determined the following formula: TX packet count Backward direction – RX packet count Backward direction = Lost Packet count Backward direction
Two-way Lost Packets (LP)	Two-way (N_{LP})	LP ≥ Two-way N_{LP}	The count of Lost Packets is determined the following formula: TX packet count Forward direction – RX packet count Backward direction = Lost Packet count Two-way

Table 9 – SOAM Performance Metrics TCA

Note that not all performance metrics are listed in Table 9. They are either not suitable or not necessary. For example:

- MPD is a performance metric measuring an average and thus a poor metric for immediate attention, compared to PD, PDR and IPDV.

If TCA functionality is supported, the following requirements are applicable for an IP SOAM PM Implementation:

[CR21]< [O4] An IP SOAM PM Implementation **MUST** support per performance metric, per PM Session configuration of TCA Functions and associated thresholds, using the parameters described in Table 9, for the following performance metrics:

- One-way IPDV in the Forward Direction
- One-way Maximum IPDV in the Forward Direction
- One-way IPDV in the Backward Direction
- One-way Maximum IPDV in the Backward Direction
- Two-way PD
- Two-way Maximum PD
- One-way PDR in the Forward Direction
- One-way Maximum PDR in the Forward Direction
- One-way PDR in the Backward Direction
- One-way Maximum PDR in the Backward Direction
- One-way PL in the Forward Direction
- One-way PL in the Backward Direction
- Two-way PL

[CR22]< [O4] If time-of-day synchronization is supported, an IP SOAM PM Implementation **MUST** support per performance metric, per PM Session configuration of TCA Functions and associated thresholds, using the parameters described in Table 9, for the following performance metrics:

- One-way PD in the Forward Direction
- One-way Maximum PD in the Forward Direction
- One-way PD in the Backward Direction
- One-way Maximum PD in the Backward direction

[CR23]< [O4] An IP SOAM PM Implementation **MUST** support stateless TCA reporting.

[CD9]< [O4] An IP SOAM PM Implementation **SHOULD** support stateful TCA reporting.

[CR24]< [O4] If an IP SOAM PM Implementation supports stateful TCA reporting, it **MUST** support a configurable parameter per TCA Function to indicate whether the TCA Function uses stateful or stateless TCA reporting.

[CR25]< [O4] An IP SOAM PM implementation **MUST** support a single configurable parameter for the threshold value for each TCA Function that uses stateless TCA reporting.

[CR26]< [O4] If an IP SOAM PM Implementation supports stateful TCA reporting, it **MUST** support the CLEAR threshold being equal to the SET threshold.

[CO1]< [O4]<[CD9]< If an IP SOAM PM Implementation supports stateful TCA reporting, it **MAY** support the CLEAR threshold being different to the SET threshold.

For thresholds defined using bins, a CLEAR threshold (N_C, k_C) is defined to be less than or equal to a SET threshold (N_S, k_S) if $k_C = k_S$ and $N_C <= N_S$.

[CR27]< [O4]<[CD9]< [CO1]< If an IP SOAM PM Implementation supports stateful TCA reporting with different SET and CLEAR thresholds, the CLEAR threshold **MUST** be less than or equal to the SET threshold.

[CR28]< [O4]<[CD9]< If an IP SOAM PM Implementation supports stateful TCA reporting, it **MUST** support a configurable parameter for the SET threshold for each TCA Function that uses stateful TCA reporting.

[CR29]< [O4]<[CD9]<[CO1]< If an IP SOAM PM Implementation supports stateful TCA reporting with different SET and CLEAR thresholds, it **MUST** support a configurable parameter for the CLEAR threshold for each TCA Function that uses stateful TCA reporting.

If different SET and CLEAR thresholds are not used, the value configured for the SET threshold is also used for the CLEAR threshold.

[CR30]< [O4] If a TCA Function is configured to use stateless TCA reporting, a TCA **MUST** be generated for each Measurement Interval in which the threshold is crossed as defined in Table 9.

[CD10]< [O4] If a TCA Function is configured to use stateless TCA reporting, the TCA for a given Measurement Interval **SHOULD** be generated as soon as the threshold is crossed.

[CR31]< [O4] If a TCA Function is configured to use stateless TCA reporting, the TCA for a given Measurement Interval **MUST** be generated within 1 minute of the end of the Measurement Interval.

[CR32]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, in the 'clear' state a SET TCA **MUST** be generated for a given Measurement Interval if the SET threshold is crossed as defined in Table 9 during that Measurement Interval.

[CR33]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, in the 'clear' state, if the SET threshold is crossed during a given Measurement Interval, the state **MUST** be changed to 'set' by the end of that Measurement Interval.

[CD11]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, the SET TCA for a given Measurement Interval **SHOULD** be generated as soon as the SET threshold is crossed.

[CR34]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, the SET TCA for a given Measurement Interval **MUST** be generated within 1 minute of the end of the Measurement Interval.

[CR35]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, SET TCAs **MUST NOT** be generated when in the 'set' state.

[CR36]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, in the 'set' state a CLEAR TCA **MUST** be generated for a given Measurement Interval if the CLEAR threshold is not crossed as defined in Table 9 during that Measurement Interval.

[CR37]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, in the 'set' state, if the CLEAR threshold is not crossed during a given Measurement Interval, the state **MUST** be changed to 'clear' at the end of that Measurement Interval.

[CD12]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, the CLEAR TCA for a given Measurement Interval **SHOULD** be generated immediately at the end of the Measurement Interval.

[CR38]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, the CLEAR TCA for a given Measurement Interval **MUST** be generated within 1 minute of the end of the Measurement Interval.

[CR39]< [O4]<[CD9]< If a TCA Function is configured to use stateful TCA reporting, CLEAR TCAs **MUST NOT** be generated when in the 'clear' state.

[CR40]< [O4] For a given TCA Function applying to a given performance metric and a given PM Session, an IP SOAM PM Implementation **MUST NOT** generate more than one TCA for each Measurement Interval.

[CR41]< [O4] An IP SOAM PM Implementation **MUST** support the configuration of at least one TCA Function for each performance metric listed in Table 6, for each PM Session.

Note: this does not require that an IP SOAM PM Implementation is able to support configuration of a TCA Function for every performance metric for every PM Session simultaneously.

[CO1]< [O4] An IP SOAM PM Implementation **MAY** support the configuration of more than one TCA Function for a performance metric, for each PM Session.

9.5.3 SOAM PM TCA Notification Messages

Table 10 lists the SOAM PM TCA Notification message attributes used when sending a TCA to an ICM/SOF.

Field Name	Field Description
Date and Time	Time of the event, in UTC. For stateless TCAs, and stateful SET TCAs, this is the time the threshold was crossed; for stateful CLEAR TCAs, it is the time at the end of the Measurement Interval for which the CLEAR TCA is being generated.
PM Session	Identification of the PM Session for which the TCA Function was configured. The specific parameters needed to uniquely identify a PM Session are implementation-specific.
Measurement Interval	The time, in UTC, at the start of the Measurement Interval for which the TCA was generated.
Performance Metric Name	Performance Metric for which the TCA Function was configured, i.e., one of those listed in Table 9.
Configured Threshold	The configured threshold parameters. For bin-based thresholds, this includes the bin number and the total count, i.e., (N, k).
Measured Performance Metric	Measured value that caused the TCA to be generated. For bin-based thresholds configured as (N, k), this is always equal to N for stateless TCAs and stateful SET TCAs; for stateful CLEAR TCAs, it is the value of UBC(k) at the end of the Measurement Interval. For "maximum" performance metrics, for stateless TCAs and stateful SET TCAs, this is the first value in the Measurement Interval that reaches or exceeds the configured threshold; for stateful CLEAR TCAs it is the maximum value at the end of the Measurement Interval.
Suspect Flag	Value of the Suspect Flag for the Measurement Interval for which the TCA was generated. Suspect Flag is true

Field Name	Field Description
	when there is a discontinuity in the performance measurements conducted during the Measurement Interval.
TCA Type	The type of TCA, i.e. one of STATELESS (if stateless TCA reporting was configured for the TCA Function), STATEFUL-SET (if stateful TCA reporting was configured and this is a SET TCA) or STATEFUL-CLEAR (if stateful TCA reporting was configured and this is a CLEAR TCA).
Severity	WARNING (for STATELESS or STATEFUL-SET) or INFO (for STATEFUL-CLEAR)

Table 10 – TCA Notification Message Fields

[CR42]< [O4] An IP SOAM PM Implementation **MUST** include the fields in the TCA notification messages listed in Table 10.

Table 11 shows the correlation between the general alarm and event notification parameters described in ITU-T X.733 [25] and X.734 [26], and the notification attributes considered in this document.

ITU-T X.733, X.734	IP Services SOAM
Event time	Date and time
Managed Obj Class	PM Session
Managed Obj Instance	Included in PM Session
Monitored Attribute	Performance Metric Name, Measurement Interval
Threshold Info	Configured Threshold, Measured Performance Metric
No Equivalent	Suspect Flag
Event Type (service degraded)	TCA Type
Severity	Severity
Probable Cause	Not applicable

Table 11 – Comparison of TCA Fields in X.73x and MEF 61

10 Hybrid Measurement

Hybrid measurement modifies the Subscriber packet in some way and uses the Subscriber packet to monitor the service rather than using synthetic packets. There are two expected benefits of using Hybrid measurement. The first is that there is no need for additional synthetic packets to be generated and carried across the network. This impacts the possibility of congestion occurring due to the addition of synthetic packets. The second is that measurement packets take the same path as Subscriber packets since the measurement packets are subscriber packets. This is true but unless every Subscriber packet is modified all possible paths that the Subscriber packets traverse might not be measured. The type of Hybrid Measurement discussed in this document is Alternate marking (AltM).

10.1 Alternate Marking Explanation

RFC 8321 [17] describes a method to perform packet loss, delay, and jitter measurements on live traffic. This method is based on an AltM (coloring) technique. This technology can be applied in various situations, and could be considered Passive or Hybrid depending on the application. The basic idea is to virtually split traffic flows into consecutive blocks and a simple way to create the blocks is to "color" the traffic. Each block represents a measurable entity unambiguously recognizable along the path and by counting the number of packets in each block and comparing the values measured by different network devices along the path it is possible to measure packet loss in any single block between any two points.

Taking into consideration RFC 7799 [15] definitions, the AltM Method could be considered Hybrid or Passive, depending on the case. In the case where the marking method is obtained by changing existing field values of the packets (e.g., the Differentiated Services Code Point (DSCP) field), the technique is Hybrid. In the case where the marking field is dedicated, reserved, and included in the protocol specification, the AltM technique can be considered as Passive (e.g., Synonymous Flow Label as described in draft-ietf-mpls-rfc6374-sfl [22] or OAM Marking Bits as described in draft-ietf-bier-pmmm-oam [18]).

Since the traffic is colored it is clear and fully identifiable within the network. If a flow is marked and counted along the path it is possible to measure not only Packet Loss and Packet Delay but IPDV can also be calculated. AltM also identifies which path the packet goes through and this enables a real time tracing of the packet. It should be noted that only the path taken by the measured packets is known, this does not mean that all packets in the flow are taking this same path.

Note: At this time the use of AltM in an IP network has not been standardized.

The basic idea of AltM is to virtually split traffic flows into consecutive blocks: each block represents a measurable entity unambiguously recognizable by all network devices along the path. By counting the number of packets in each block and comparing the values measured by different network devices along the path, it is possible to measure packet loss occurred in any single block between any two points. The simplest way to create the blocks is to "color" the traffic e.g. setting proper values for one or two bits (two colors are sufficient), so that packets belonging to different consecutive blocks will have different colors. Whenever the color changes, the previous block terminates and the new one begins. Hence, all the packets belonging to the same block will have the same color and packets of different consecutive blocks will have different colors. Figure 19 shows a representation of the AltM methodology.

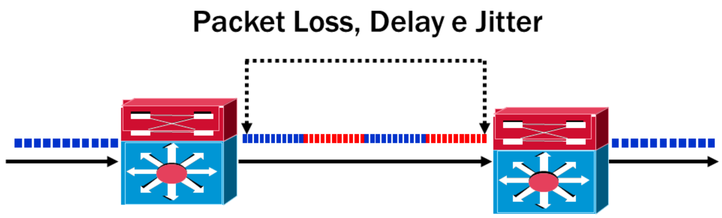

Figure 19 – AltM description

There are two alternatives for color switching: using a fixed number of packets or a fixed timer. However, using a fixed timer for color switching offers better control over the method. The time length of the blocks can be chosen large enough to simplify the collection and the comparison of measurements taken by different network devices.

In addition, two different strategies can be used when implementing the method: link-based and flow-based. The end-to-end measurement can be split into Hop-by-Hop measurements (for each Link and/or each Router).

The flow-based strategy is used when only a part of all the traffic flows in the operational network need to be monitored. According to this strategy, only a subset of the flows is colored. Counters for packet loss measurements can be instantiated for each single flow, or for the set as a whole, depending on the desired granularity. Router1, Router2,… RouterN are configured to have dedicated counters for the different flows under monitoring.

The link-based measurement is performed on all the traffic on a point to point link-by-link basis. The link could be a physical link or a logical link. Counters could be instantiated for the traffic as a whole without distinction of the flow. Router1, Router2,… RouterN are not configured to filter any flow.

So, in order to perform the desired performance measurement for Subscriber's IP Service from PE to PE, the flow-based strategy can be used and the interested flows can be selected based on Subscriber's IP addresses. Both End-to-End and Hop by Hop measurements can be applied depending on the necessity.

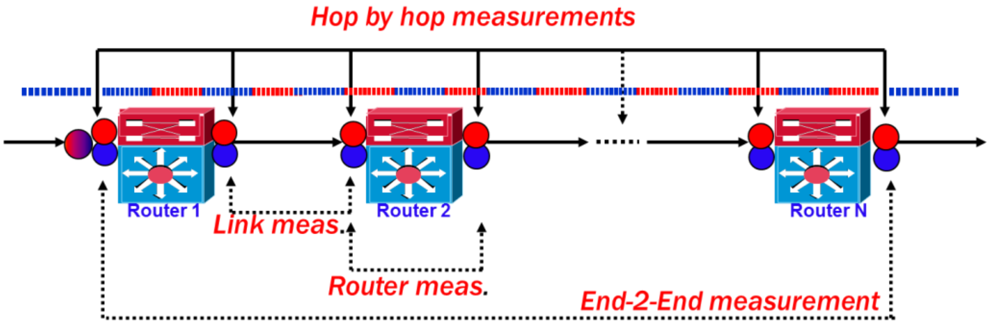

Figure 20 – AltM measurement strategies

It is possible to have Hop by hop measurements (Link meas. and Router meas.) or only End-to-End measurement depending on the case. If the IP service from PE to PE is MPLS based, Hop by hop measurements cannot be performed while End-to-End measurement is allowed.

Since a Service Provider application is described here, the method can be applied to End-to-End services supplied to Customers and the method should be transparent outside the PM domain. So the source node (e.g. Router 1 that can be a PE) marks the packets while the destination node (e.g. Router N that can be another PE) could restore the marking value to the initial value depending on the implementation.

The same principle used to measure packet loss can be applied also to one-way delay measurement. Note that, for all the one-way delay alternatives described, by summing the one-way delays of the two directions of a path, it is always possible to measure the two-way delay (round-trip "virtual" delay). The limitation with measuring two-way delay is that the one-way measurements are based on Subscriber packets. It is very likely that a Subscriber will send more packets in one direction than in the other which means that there will be more one-way delay measurements in one direction than the other. The two-way delay measurement would be an approximation at best.

10.1.1 Single-Marking Methodology

The alternation of colors can be used as a time reference to calculate the delay. A measurement is valid only if no packet loss occurs and if packet misordering can be avoided.

10.1.2 Mean Delay

A different approach can be considered in order to overcome the sensitivity to out-of-order: it is based on the concept of mean delay. The mean delay is calculated by considering the average arrival time of the packets within a single block. The network device locally stores a timestamp for each packet received within a single block: summing all the timestamps and dividing by the total number of packets received, the average arrival time for that block of packets can be calculated. By subtracting the average arrival times of two adjacent devices, it is possible to calculate the mean delay between those nodes. This method is robust to out-of-order packets and also to packet loss (only a small error is introduced).

10.1.3 Double-Marking Methodology

The limitation of mean delay is that it doesn't give information about the delay value's distribution for the duration of the block. Additionally, it may be useful to have not only the mean delay but also the minimum, maximum, and median delay values and, in wider terms, to know more about the statistic distribution of delay values. So, in order to have more information about the delay and to overcome out-of-order issues, a different approach can be introduced; it is based on a Double-Marking methodology.

Basically, the idea is to use the first marking to create the alternate flow and, within this colored flow, a second marking to select the packets for measuring delay/jitter. The first marking is needed for packet loss and mean delay measurement. The second marking creates a new set of marked packets that are fully identified over the network, so that a network device can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on a second router (the double marked packets in the same order) to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. The frequency of the second marking must not be too high in order to avoid out-of-order issues. For example if the time length of the blocks is short (e.g. 100ms) only one double marked packet should be inserted. If the time length of the blocks is longer (e.g. 10 s) more double marked packets in a single block could be inserted, with a gap time between two of them big enough to avoid out of order packets. With the right gap time between consecutive double marked packets, the order of these packets will remain the same.

Similar to one-way delay measurement (both for Single Marking and Double Marking), the method can also be used to measure the IPDV.

The latest developments of RFC 8321 [17] are described in draft-fioccola-ippm-multipoint-alt-mark [19] that generalizes AltM technology to multipoint-to-multipoint scenario. The idea is to expand Performance Management methodologies to measure any kind of unicast flows, also multipoint-to-multipoint, where a lot of flows and nodes have to be monitored. This is very useful for a Performance Management SDN Controller Application.

10.2 Alternate Marking for FM

The main target for AltM is PM. The use of AltM for Proactive and On-demand Fault Monitoring has been proposed but not standardized. It might be possible to trace the path of a given flow through the network.

Since the traffic is marked, it is recognizable by all network devices along the path that can identify the marking and the flow tracing can be enabled. As stated previously, if the core network is an MPLS network, it is not possible to trace IP packets through the MPLS network.

10.3 Alternate Marking for PM

AltM can provide the ability to measure the performance of a service through the use of its coloring techniques. Measurements such as PD and PL are possible using AltM.

IETF Working Draft draft-mizrahi-ippm-compact-alternate-marking provides a summary of all the AltM method alternatives. Specific methods have not been adopted.

11 References

[1] IETF RFC 791, *Internet Protocol, DARPA Internet Program Protocol Specification*, September 1981

[2] IETF RFC 792, *Internet Control Message Protocol*, September 1981

[3] IETF RFC 1321, *The MD5 Message Digest Algorithm*, April 1992

[4] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, March 1997

[5] IETF RFC 3031, *Multiprotocol Label Switching Architecture*, January 2001

[6] IETF RFC 3174, *US Secure Hash Algorithm 1 (SHA1)*, September 2001

[7] IETF RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*, February 2006

[8] IETF RFC 4443, *Internet Control Message Protocol (IPv6) for the Internet Protocol Version 6 (IPv6) Specification*, March 2006

[9] IETF RFC 4656, *A One-way Active Measurement Protocol (OWAMP)*, September 2006

[10] IETF RFC 5357, *Two-Way Active Measurement Protocol (TWAMP)*, October 2008

[11] IETF RFC 5880, *Bidirectional Forwarding Detection*, June 2010

[12] IETF RFC 5881, *Bidirectional Forwarding Detection for IPv4 and IPv6 (Single Hop)*, June 2010

[13] IETF RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*, June 2010

[14] IETF RFC 7419, *Common Interval Support in Bidirectional Forwarding Detection*, December 2014

[15] IETF RFC 7799, *Active and Passive Metrics and Methods (with Hybrid Types In-Between)*, May 2016

[16] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, May 2017

[17] IETF RFC 8321, *Alternate Marking Method for Passive and Hybrid Performance Monitoring*, January 2018

[18] IETF Working Draft draft-ietf-bier-pmmm-oam, *Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer*, June 2018

Editor Note 2: *The above working draft is expected to be finalized by March 2019. If this is not finalized by the time that this document goes to Letter Ballot we will remove the references to this document.*

[19] IETF Working Draft draft-fioccola-ippm-multipoint-alt-mark, *Multipoint Alternate Marking method for passive and hybrid performance monitoring*, June 2017

Editor Note 3: The above working draft is expected to be finalized in September 2019. If this is not finalized by the time that this document goes to Letter Ballot we will remove the references to this document.

[20] IETF Working Draft draft-ietf-ippm-stamp, *Simple Two-way Active Measurement Protocol (STAMP)*, 03/21/2018

Editor Note 4: The above working draft is expected to be finalized in June 2019. If this is not finalized by the time that this document goes to Letter Ballot we will remove the references to this document.

[21] IETF Working Draft draft-ietf-ippm-stamp-yang, *Simple Two-way Active Measurement Protocol (STAMP) Data Model*, 03/01/2018

Editor Note 5: The above working draft is expected to be finalized in June 2019. If this is not finalized by the time that this document goes to Letter Ballot we will remove the references to this document.

[22] IETF Working Draft draft-ietf-mpls-rfc6374-sfl, RFC6374 *Synonymous Flow Labels*, June 2017

Editor Note 6: The above working draft is expected to be finalized in January 2020. If this is not finalized by the time that this document goes to Letter Ballot we will remove the references to this document.

[23] ISO 8601, *Data elements and interchange formats –Information interchange -- Representation of dates and times*, 2004

[24] ITU-T Recommendation G.7710/Y.1701, *Common Equipment Management Function Requirements*, February 2012, November 2016

[25] ITU-T Recommendation X.733, *Information Technology – Open Systems Interconnection – Systems Management: Alarm Reporting Function*, February 1992, February 1994, April 1995, October 1996, March 1999

[26] ITU-T Recommendation X.734, *Information Technology – Open Systems Interconnection – Systems Management: Event Report Management Function*, September 1992, February 1994, April 1995, October 1996, March 1999

[27] MEF 10.4, *Ethernet Services Attributes Phase 4*, xxx 2019

[28] MEF 15, *Requirements for Management of Metro Ethernet Network Phase 1 Network Elements*, November 2005

[29] MEF 17, *Service OAM Requirements and Framework - Phase 1*, April 2007

[30] MEF 30.1, *Service OAM Fault Management Implementation Agreement: Phase 2*, April 2013

[31] MEF 35.1, *Service OAM Performance Monitoring Implementation Agreement*, May 2015

[32] MEF 55, *Lifecycle Service Orchestration (LSO): Reference Architecture and Framework,* March 2016

[33] MEF 61.1, *IP Service Attributes for Subscriber IP Services*, xxx 2019

[34] Telcordia GR-253-CORE, *SONET Transport Systems: Common Criteria*, September 2000

Appendix A Life Cycle Terminology (Informative)

The following diagrams show how the life cycle terminology (see section9.2.1) for a PM Session is used in this document. While measurements are being taken for a PM Session, the Message Period specifies the time interval between IP SOAM Measurement packets, and therefore how often the IP SOAM Measurement packets are being sent. The Measurement Interval is the amount of time over which the statistics are collected and stored separately from statistics of other time intervals.

Each PM Session supports Single-ended Delay and Single-ended PL measurements for a specific IP CoS Name on a specific Pair of MPs.

A PM Session can be Proactive or On-Demand. While there are similarities, there are important differences and different attributes for each. Each is discussed below in turn.

A.1 Proactive PM Sessions

For a Proactive PM Session, there is a time at which the session is created, and the session may be deleted later. Other attributes include the Message Period, Measurement Interval, Repetition Period, Start Time (which is always ‘immediate’ for Proactive PM Sessions), and Stop Time (which is always ‘forever’ for Proactive PM Sessions).

The IP SOAM Measurement packets associated with the PM Session are transmitted every “Message Period”. Data in the form of counters is collected during a Measurement Interval (nominally 15 minutes) and stored in a Current data set. When time progresses past the Measurement Interval, the former Current data set is identified as a History data set. There are multiple History data sets, and the oldest is overwritten.

The SOF/ICM will combine the counters retrieved from devices or virtual applications to calculate estimates over the SLS period T.

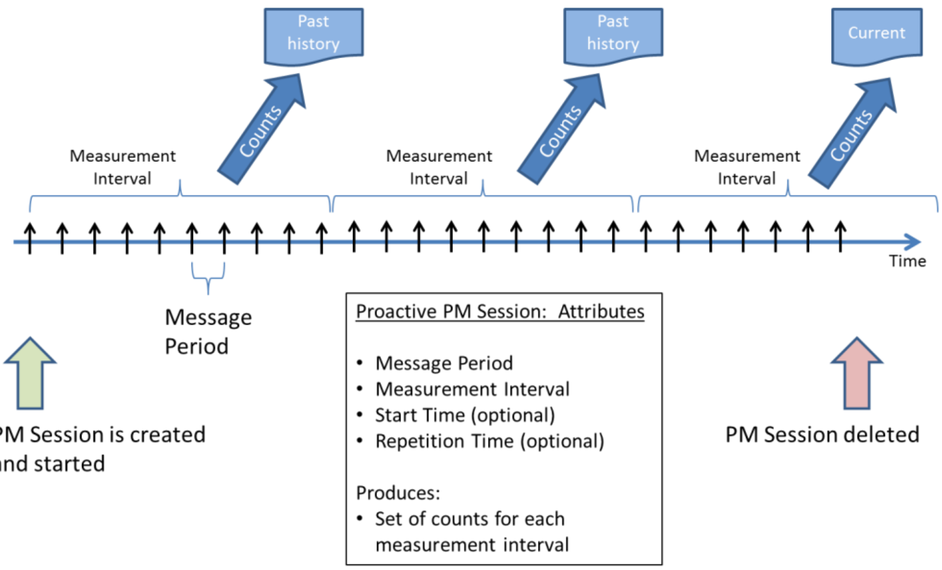

Figure 21 – Measurement Interval Terminology

A.2 On-Demand PM Sessions

For On-Demand PM Sessions, there is a Start Time and a Stop Time. Other attributes can include Message Period, Measurement Interval, and Repetition Time, depending on the type of session that is requested. Different examples are shown in the subsequent diagrams.

Note, in all examples it is assumed that during the interval data is being collected for a report, the counters of the report do not wrap. This is affected by the frequency IP SOAM Measurement packets are sent, the length of time they are sent, and the size of the report counters; the details are not addressed in this specification. At least one report is assumed to be saved after the Measurement Interval is complete.

In the first example, the On-Demand session is run and one set of data is collected. That is, in this example, multiple Measurement Intervals are not used.

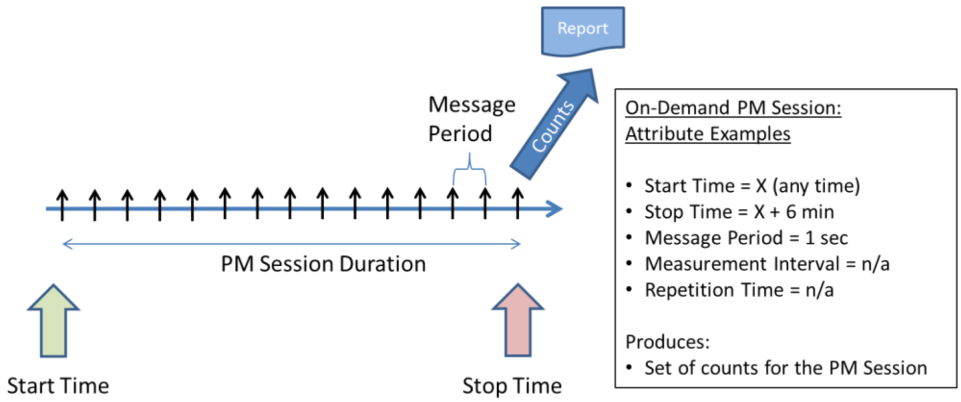

Figure 22 – Illustration of non-Repetitive, On-Demand PM Session

On-Demand PM Sessions can be specified so that Repetitions are specified. This is shown below. Note that a report is created at the end of each Measurement Interval (or Stop Time, if that occurs before the end of the Measurement Interval).

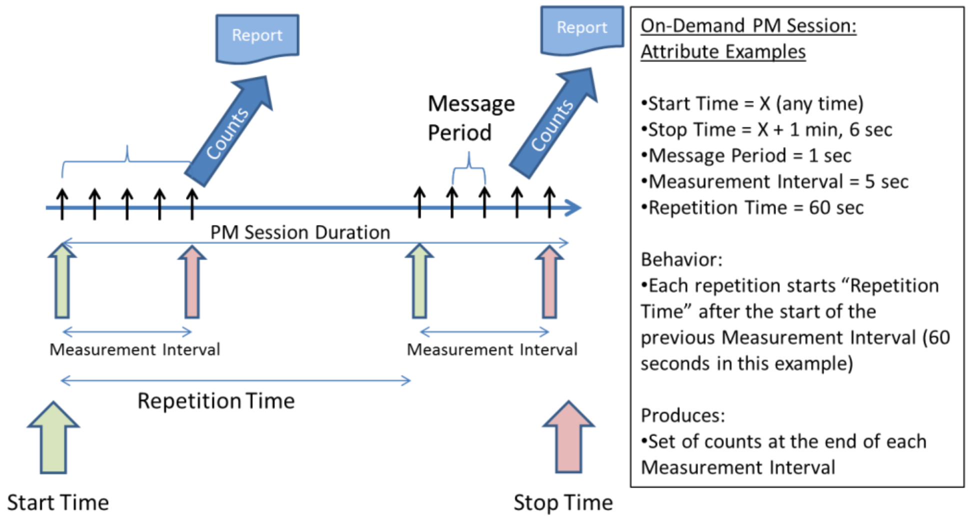

Figure 23 – Example of Repetitive On-Demand PM Session

A.3 PM Sessions With Clock-Aligned Measurement Intervals and Repetition Time of "None"

In all of the previous examples, Measurement Intervals were aligned with the PM Session, so that a PM Session Start Time always occurred at the beginning of a Measurement Interval. Measurement Intervals can instead be aligned to a clock, such as a local time-of-day clock.

When Measurement Intervals are aligned to a clock, then in general the PM Session Start Time will not coincide with the beginning of a Measurement Interval.

When the Repetition Time is “none”, then the PM Session Start Time will always fall inside a Measurement Interval, so measurements will begin to be taken at the Start Time. As Figure 24 illustrates, when Measurement Intervals are aligned with a clock rather than aligned with the PM Session, then the first Measurement Interval could be truncated. The first, truncated Measurement Interval ends when the clock-aligned Measurement Interval boundary is reached. If the PM Session is Proactive, then a report is generated as usual, except that this report will have the Suspect Flag set to indicate the Measurement Interval’s truncated status. Figure 24 depicts a Proactive PM Session, but the same principles apply to On-Demand PM Sessions with Repetition Times of “none”.

Subsequent Measurement Intervals in the PM Session will be of full length, with Measurement Interval boundaries occurring at regular fixed-length periods, aligned to the clock. The exception may be the last Measurement Interval of the PM Session. When a PM Session is Stopped or Deleted, then the final Measurement Interval could be truncated, and so again the Suspect Flag would be set for this final, truncated Measurement Interval.

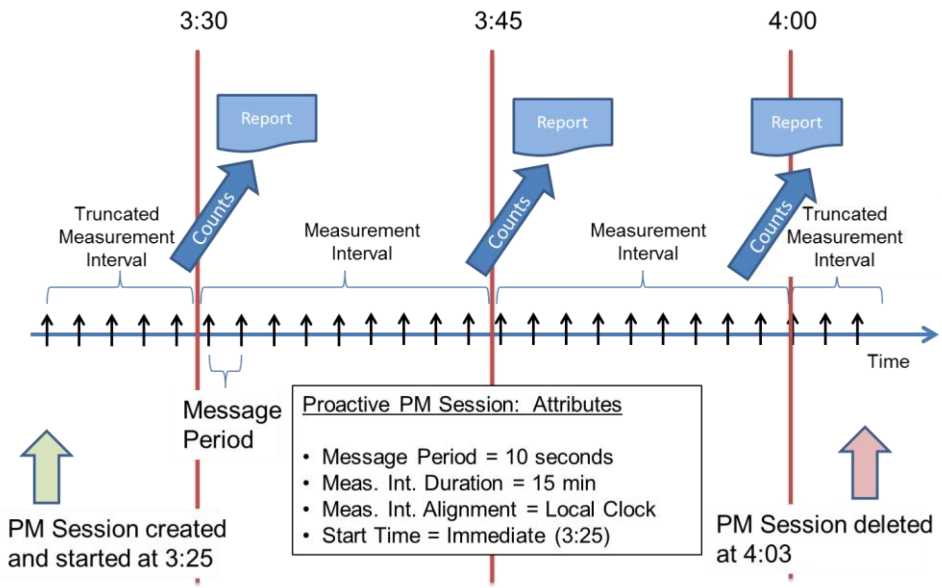

Figure 24 – Example Proactive PM Session with Clock-Aligned Measurement Interval

A.4 PM Sessions With Clock-Aligned Measurement Intervals and Repetition Times Not Equal To “None”

When Measurement Intervals are aligned with a clock and the Repetition Time is not equal to “none”, then there are two possibilities for the PM Session Start Time. The first possibility is that the PM Session Start Time is at a time that would fall inside a clock-aligned Measurement Interval. The second possibility when Repetition Times are not equal to "none" is that the PM Session Start Time could fall outside of a clock-aligned Measurement Interval.

If the PM Session Start Time would fall inside a clock-aligned Measurement Interval, then measurements would begin immediately at the PM Session Start Time. In this case, the first Measurement Interval might be truncated (unless PM Session Start Time is also chosen to align with local clock), and thus have its data flagged with a Suspect Flag. An example is illustrated in Figure 25. Figure 25 depicts an On-Demand PM Session, but the same principles apply to a Pro-active PM Session whose Repetition Time is not equal to "none".

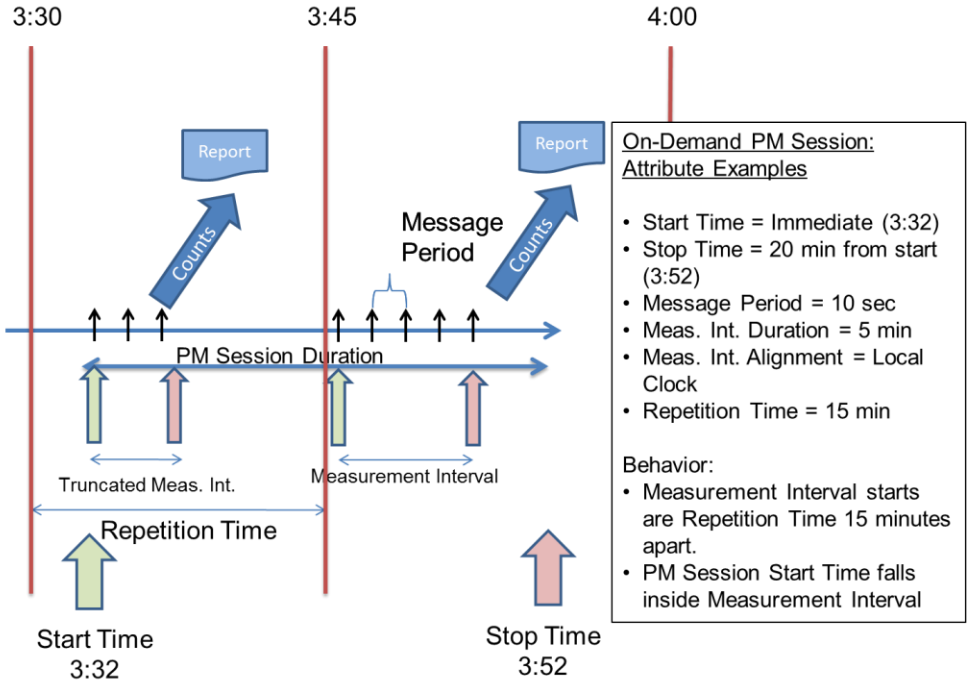

Figure 25 – Example On-Demand PM Session with Clock-Aligned Measurement Interval

In Figure 25, the PM Session starts at 3:32 and has a Stop Time at 3:52. Note that the PM Session might not have been given these explicit times; the PM Session could have had a Start Time of "immediate" and a Stop Time of "20 minutes from start". The Measurement Interval boundary is aligned to the local clock at quadrants of the hour. The next Measurement Interval boundary after the PM Session Start Time is at 3:45. Since the Repetition Time is 15 minutes and the Measurement Interval duration is 5 minutes, the PM Start Time of 3:32 falls inside a Measurement Interval, therefore measurements are begun at the PM Start Time. The first Measurement Interval ends at 3:35 due to its alignment with the local clock. Therefore, the first Measurement Interval is a truncated Measurement Interval (3 minutes long rather than the normal 5 minutes) and its data will be flagged with the Suspect Flag.

The next Measurement Interval begins at 3:45, and runs for its full 5 minute duration, so measurements cease at 3:50. In this example, the PM Session reaches its Stop Time before any more Measurement Intervals can begin. Note that the PM Session Stop Time could fall inside a Measurement Interval, in which case the final Measurement Interval would be truncated; or the PM Session could fall outside a Measurement Interval, in which case the final Measurement Interval would not be truncated. In Figure 26, the data from the second Measurement Interval would not be flagged as suspect.

Figure 25 covered the case where the PM Session Start Time falls inside a clock-aligned Measurement Interval. The second possibility when Repetition Times are not equal to "none" is that the PM Session Start Time could fall outside of a clock-aligned Measurement Interval. In such a case, measurements would not begin immediately at the PM Session Start Time, but rather would be delayed until the next Measurement Interval begins. An example is illustrated in Figure 26. Again, while Figure 26 depicts an On-Demand PM Session, similar principles apply to a Proactive PM Session whose Repetition Time is not equal to "none".

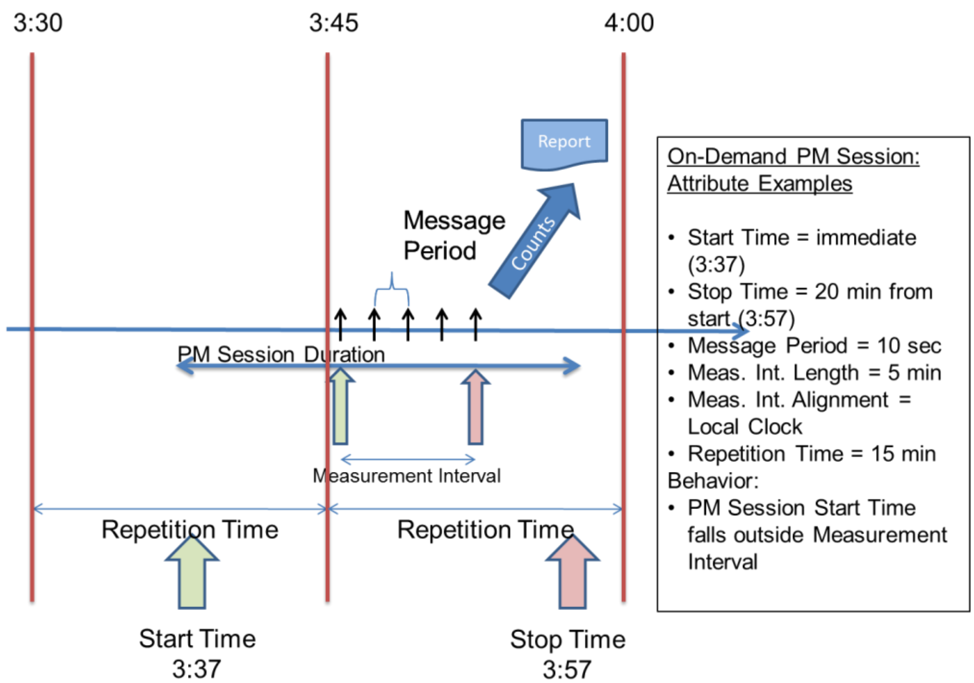

Figure 26 – Second Example of On-Demand PM Session with Clock-Aligned Measurement Interval

In Figure 26, the PM Session starts at 3:37 and has a Stop Time at 3:57. Note that the PM Session might not have been given these explicit times; the PM Session could have had a Start Time of "immediate" and a Stop Time of "20 minutes from start". Note also that in such a case, the parameters given in Figure 26 might be identical to the parameters given in Figure 25, with the only difference being that the “Start button” is pressed 5 minutes later.

The Measurement Interval boundary is aligned to the local clock at quadrants of the hour. The next Measurement Interval boundary after the PM Session Start Time is at 3:45. Since the Repetition Time is 15 minutes and the Measurement Interval duration is 5 minutes, the PM Start Time of 3:37 falls outside a Measurement Interval. Therefore, measurements do not begin at the PM Session Start Time but instead are delayed until the next Measurement Interval boundary.

The first Measurement Interval for this example begins at 3:45, 8 minutes after the PM Session is started. This first Measurement Interval runs for its full 5 minutes, so its data will not have the Suspect Flag set. Measurements cease at 3:50 due to the 5 minute Measurement Interval duration. In this example, the PM Session reaches its Stop Time before any more Measurement Intervals can begin.

Note that, as in the previous case, the PM Session Stop Time could fall either inside or outside a Measurement Interval, and so the final Measurement Interval might or might not be truncated. In general, all Measurement Intervals other than the first and last Measurement Intervals should be full-length.

Appendix B Measurement Bins (Informative)

MEF 61.1 [33] performance metrics of One-way Packet Delay Performance, One-way Packet Delay Range, and Inter-Packet Delay Variation Performance are all defined in terms of the p-Percentile of packet delay or inter-packet delay variation. Direct computation of percentiles would be resource intensive, requiring significant storage and computation. This informative appendix describes a method for determining whether performance objectives are met using bins for packet delay, inter-packet delay variation, and packet delay range.

B.1 Description of Measurement Bins

As described in section 9.5.1.2, each packet delay bin is one of n counters, B1, .. Bn, each of which counts the number of packet delay measurements whose measured delay, x, falls into a range. The range for n+1 bins (there are n bins, plus Bin 0, so n+1) is determined by n delay thresholds, D1, D2, .. Dn such that 0 < D1 < D2 < .. < Dn. Then a packet whose delay is x falls into one of the following delay bins:

Bin *0* if $x < D_1$

Bin *i* if $D_i \leq x < D_{i+1}$

Bin *n* if $Dn \leq x$

Note: A Bin *0* ($\mathbf{B_0}$) counter does not need to be implemented, because, B_0 can be determined from R, the total number of IP SOAM Measurement packets received using the following formula:

$$B0 = R - \sum_{i=1}^{n} Bi$$

Similarly, each inter-packet delay variation (IPDV) bin is one of m counters, B1, … ,Bm, each of which counts the number of IPDV measurements whose measured delay, v falls into a range. The range for m+1 bins is determined by m IPDV thresholds, V1, V2, .. Vm such that 0 < V1 < V2 < .. < Vm. Then a packet whose IPDV v falls into one of the following IPDV bin:

Bin 0 if $v < V_1$

Bin *i* if $V_i \leq v < V_{i+1}$

Bin m if $V_m \leq x$

Note: A Bin *0* (B_0) counter does not need to be implemented, because B0 can be determined from Ry, the total number of IPDV measurement packet pairs received using the following formula:

$$B0 = Ry - \sum_{i=1}^{m} Bi$$

B.2 One-way Packet Delay Performance

As defined in MEF 61.1 the One-way Packet Delay Performance is met for an Pair of MPs if $Pp(x) < D$ where $Pp(x)$ is the *pth* percentile of One-Way packet delay, x and D is the One-Way packet delay performance objective set for that Pair of MPs. To determine if this objective is met, assume that of the n delay bins defined for the Pair of MPs bin j is defined such that $Dj = D$.

Then we can conclude:

$$Pp\,(x) < D \text{ if and only if } \quad \sum_{i=j}^{n} Bi < (1-p)R$$

For example, consider an objective for a Pair of MPs that the 95th percentile of One-way delay must be less than 2 milliseconds. If fewer than 5 out of 100 of the received packets have delay greater than 2 milliseconds, then the 95th percentile of delay must be less than 2 milliseconds.

B.3 One-way Inter Packet Delay Performance

As defined in MEF 61.1 [33] the One-way Inter-Packet Delay Variation Performance is met for an Pair of MPs if $Pp(v) < V$ where $Pp(v)$ is the *p*th percentile of One-way IPDV, v and V is the One-way IPDV performance objective set for that Pair of MPs. To determine if this objective is met, assume that of the m IPDV bins defined for the Pair of MPs, bin j is defined such that $Vj = V$

Then we can conclude:

$$Pp(v) < V \text{ } if \text{ } and \text{ } only \text{ } if \sum_{i=j}^{m} Bi < (1-p)Ry$$

B.4 One-way Packet Delay Range Performance

As defined in MEF 61.1 [33] the One-way Packet Delay Range Performance is met for an Pair of MPs if $Q_h(x) = P_h(x) - P_0(x) < Q$ where x is the One-way packet delay, h is a high percentile such that $0 < h \leq 1$, $P_0(x)$ is the 0^{th} percentile (i.e., the minimum) of One-way packet delay and the lower bound of the range, $P_h(x)$ is the h^{th} percentile of One-way packet delay and the higher bound of the range, and Q is the One-way packet delay range performance objective for that Pair of MPs. When $h = 1$ then $P_h(x) = \text{maximum}(x)$.

Note that requirements for measurements of minimum and maximum One-way delay are found in section 9.2. Also note that the minimum delay is lower bounded by c, the propagation delay of the shortest path connecting the Pair of MPs. The constant c could be known when the IPVC is designed.

There are two cases to consider, depending on the value of h.

B.4.1 Case 1: $Q_1(x)$

In the case where $h = 1$ then by definition $Q_1(x) = \max(x) - \min(x)$ and bins are not required to determine if the range objective is met:

$$Q1(x) < Q \ if \ and \ only \ if \ \max(x) - \min(x) < Q$$

B.4.2 Case 2: $Q_h(x)$

In the case where $h < 1$ then to determine if the objective is met, assume that of the n delay bins defined for the Pair of MPs, bin j is defined such that $D_j = c+Q$. Then we can transform the range attribute being met into a test that the upper bound on the range $P_h(x)$ is less than a known value, D_j and that the lower bound is above a known value, c, then the range will be less than their separation Q. The Equation above for One-way Packet Delay gives us a way to determine if the upper bound is less than a known value:

$$Ph(x) < Dj \ if \ and \ only \ if \ \sum_{i=j}^{n} Bi < (1-h)R$$

And so we can conclude:

$$if \ \sum_{i=j}^{n} Bi < (1-h)R \ and \ c < \min(x) \ then \ Qh(x) < Q$$

In other words, the measured range $Q_h(x)$ is less than the objective Q, and so the range objective is met.

Appendix C Statistical Considerations for Loss Measurement (Informative)

This appendix provides considerations on how to configure the Measurement Interval and Measurement Period of the Loss Measurement capability. Measurement of Packet Loss is performed using IP SOAM PM Data packets. These are not Subscriber data packets but instead they are Synthetic data packets used specifically to measure the performance of an IP service. In the sections below, where the term Synthetic packets is used, this refers to IP SOAM Data packets.

C.1 Synthetic Packets and Statistical Methods

One of the first questions of statistical analysis is, “what is the required confidence interval?” This is a central question when one is comparing a null hypothesis against an alternate hypothesis, but for this problem, it is not immediately clear what the null hypothesis is.

The assumption is that if we are promising a loss rate of alpha% to a customer, we have to build the network to a slightly smaller loss rate (otherwise, any measurement, no matter how large and accurate the sample size, would yield violations half of the time). As an example, suppose a carrier promises a network with better than 1% loss, and builds a network to .7% loss. The carrier can then choose a one-tailed confidence interval (say 95%), and then it becomes straightforward to calculate the number of samples that are needed to get the variability of measurements to be as small as needed. This is shown below.

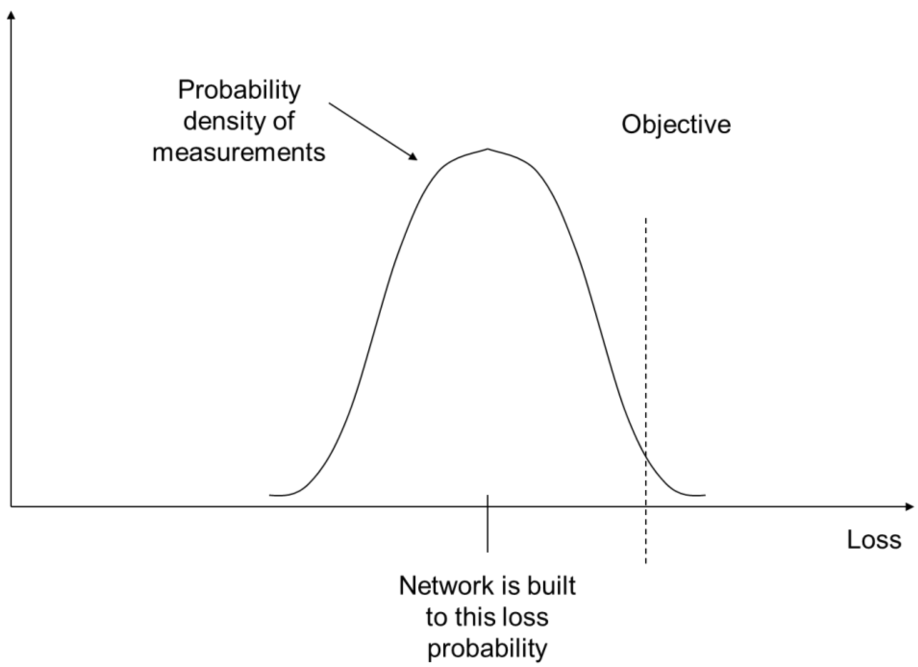

Figure 27 – Hypothesis Test for Synthetic Packet Loss Measurements

Before we specify confidence intervals, or decide how much "better" the network should be built than promised, we can study how the sampling rate and sampling interval relate to the variability of measurements. A useful measure is the Coefficient of Variation (CoV), i.e. the ratio of a probability density's standard deviation to its mean. In the hypothetical diagram above, the value would be roughly 0.2. It should be clear that the smaller the CoV, the more accurate the measurements will be.

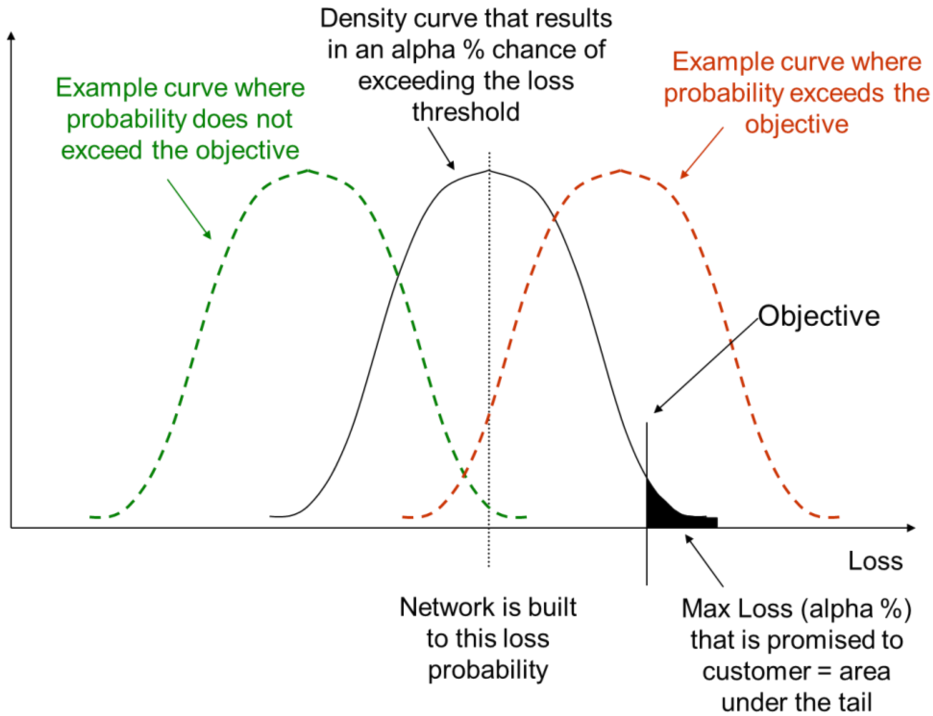

Figure 28 – Density Curve and Probability of Exceeding the Objective

Before getting into the simple equations that are relevant to the analysis, consider what the graphs look like for the Synthetic Packet approach, with specific examples of different Synthetic Packet Message Periods, Measurement Intervals, and probabilities of loss (i.e., the true Packet Loss Ratio of the network). These graphs are not hypothetical; they use exact values from the binomial probability density function. The assumption here is that the network is performing at exactly the PLR listed in the title of each graph, and the Y axis shows the probability that a specific percentage of Synthetic Packets would be lost in practice, i.e., that the measured PLR has the value shown on the X axis. Note that for some combinations of variables, the distribution is quite asymmetric with a long tail to the right, but for many others the distribution is an extremely close approximation to the normal. This, of course, is a well-known property of the binomial density function.

In each example, the number of samples (i.e., the number of Synthetic Packets) is shown - this is a function of the Message Period and the interval over which the PLR is calculated. For instance, sending one Synthetic Packet per second for 1 hour yields 3600 samples.

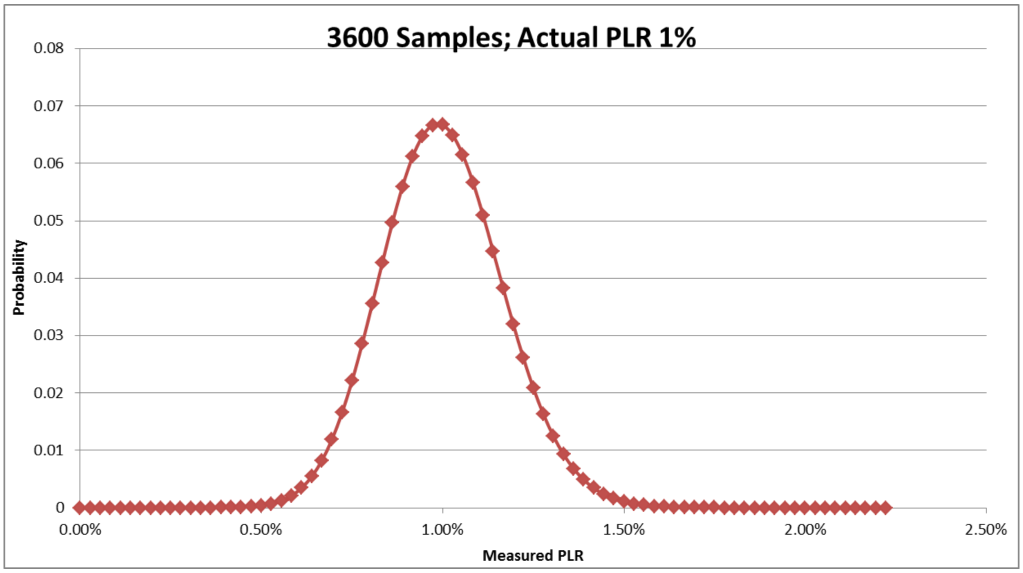

Figure 29 – Synthetic Loss Performance Example 1

The above has a CoV of 0.17. Note how it looks like a normal density.

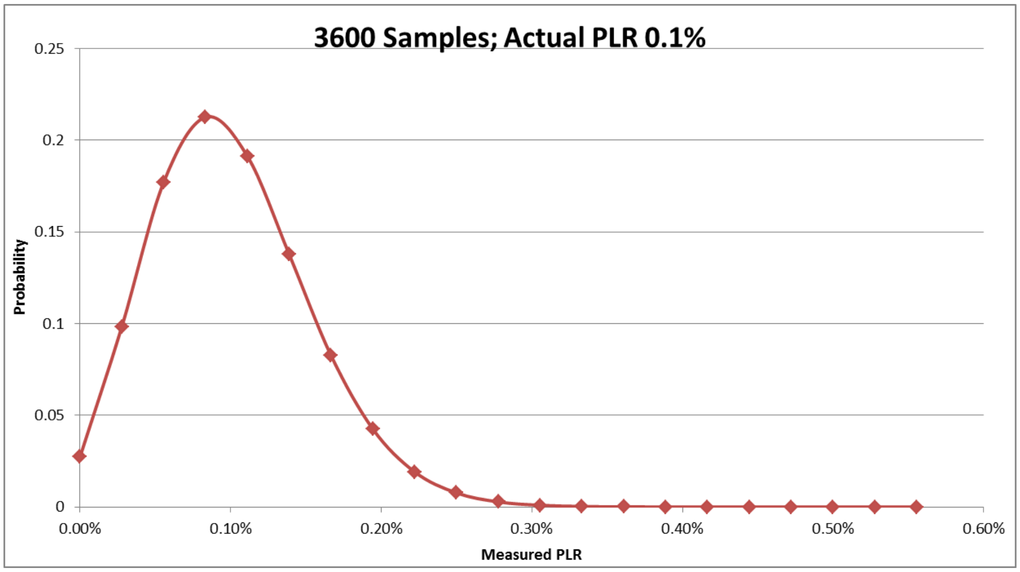

Figure 30 – Synthetic Loss Performance Example 2

In Example 2, the loss rate is smaller, and the CoV is 0.53. This is asymmetric, and variability seems too large for our use.

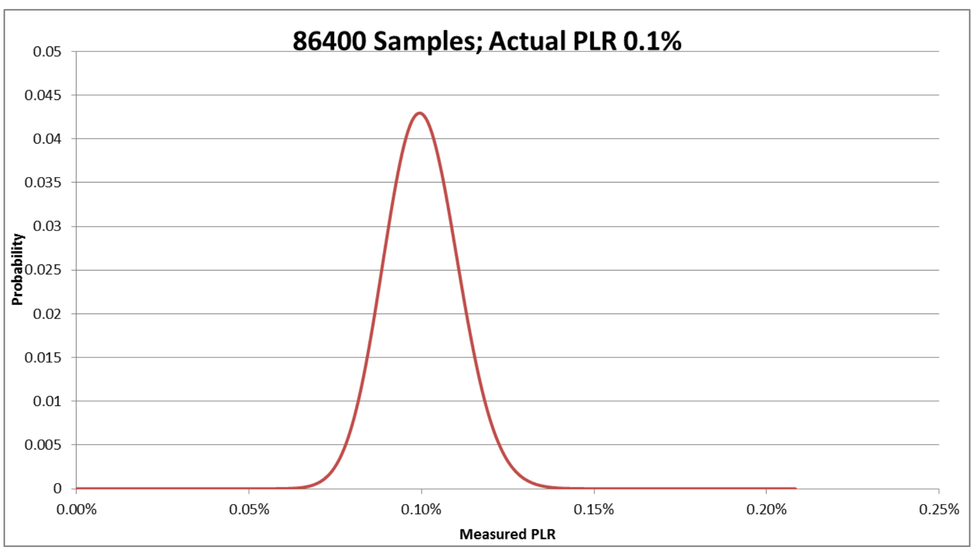

Figure 31 – Synthetic Loss Performance Example 3

Example 3 is the same as Example 2, but with a larger Measurement Interval and hence a higher number of samples. It has a CoV of 0.11 and appears to be precise enough for use.

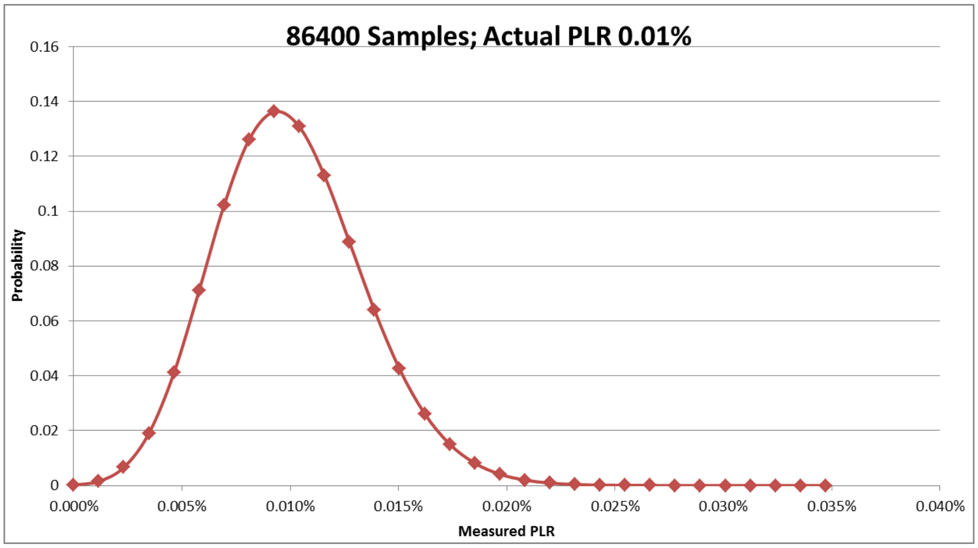

Figure 32 – Synthetic Loss Performance Example 4

In Example 4, the loss rate is even smaller. It has a CoV of 0.34, and may be too variable. Some similarities in patterns are clear; for example as the probability of packet loss (*p*) gets smaller, the effects can be mitigated by having a larger number of synthetic loss packets (*n*). This is predicted by fundamental properties of the density function. The binomial approximates the normal distribution for most of the types of numbers of concern. The exceptions are when the CoV is poor as shown in Examples 2 and 4.

The statistical properties are such that the following equations apply, where *p*=probability that a packet is lost, *q=1-p* is the probability that a packet is not lost and *n* is the sample size:

Expected number of packet lost (i.e., mean) $= \mu n = np$

Standard deviation of number of packets lost $= \sigma n = \sqrt{npq}$

These can be easily converted into PLRs:

Expected measured PLR (i.e., mean)= $\mu PLR = \frac{\mu n}{n}$ = p

Standard deviation of measured PLR = $\sigma PLR = \frac{\sigma n}{n} = \sqrt{\frac{pq}{n}}$

Note that the expected value of the measured PLR (μ_{PLR}) is always equal to the probability of loss (*p*), i.e., the actual PLR of the network.

As introduced above, the coefficient of variation, of the sample statistic is the standard deviation as a fraction of the mean:

$$\frac{\sigma}{\mu} = \frac{\sqrt{npq}}{np} = \sqrt{\frac{q}{np}} = \sqrt{\frac{q}{p}} * \frac{1}{\sqrt{n}}$$

This is the key result. The smaller CoV is, the better. For a given CoV, we can state the following:

- As *n* goes up by a factor of 10, the CoV gets smaller (improves) by a factor of $\frac{1}{\sqrt{10}}$, or about 1/3.
- As *n* goes down by a factor of 10, the CoV gets larger (gets worse) by a factor of $\sqrt{10}$, or about 3.

Furthermore, if p goes down by a certain factor, then n needs to go up by the same factor. That is, if we need to support a loss probability that is 1/100th of what we comfortably support today, we have to either increase the rate of Synthetic Packets by 100 if we sample over the same interval, increase the interval by a factor of 100, or some combination of the two such as increasing both the rate and the interval by a factor of 10.

Below are example calculations of the Coefficient of Variation. Values are highlighted where the CoV is less than 0.2. This value is proposed as a reasonable bound.

	n	*p*	μ_{PLR}	σ_{PLR}	CoV
1 hour	3600	0.01	1.000%	0.1658%	0.1658
	3600	0.001	0.100%	0.0527%	0.5268
	3600	0.0001	0.010%	0.0167%	1.6666
	3600	0.00001	0.001%	0.0053%	5.2704
24 hour	86400	0.01	1.000%	0.0339%	0.0339
	86400	0.001	0.100%	0.0108%	0.1075
	86400	0.0001	0.010%	0.0034%	0.3402
	86400	0.00001	0.001%	0.0011%	1.0758
1 month	2592000	0.01	1.000%	0.0062%	0.0062
	2592000	0.001	0.100%	0.0020%	0.0196
	2592000	0.0001	0.010%	0.0006%	0.0621

	2592000	0.00001	0.001%	0.0002%	0.1964

Table 12 – CoV Calculations with Message Period 1s

	n	p	μ_{PLR}	σ_{PLR}	CoV
1 hour	36000	0.01	1.000%	0.0524%	0.0524
	36000	0.001	0.100%	0.0167%	0.1666
	36000	0.0001	0.010%	0.0053%	0.5270
	36000	0.00001	0.001%	0.0017%	1.6667
24 hour	864000	0.01	1.000%	0.0107%	0.0107
	864000	0.001	0.100%	0.0034%	0.0340
	864000	0.0001	0.010%	0.0011%	0.1076
	864000	0.00001	0.001%	0.0003%	0.3402
1 month	25920000	0.01	1.000%	0.0020%	0.0020
	25920000	0.001	0.100%	0.0006%	0.0062
	25920000	0.0001	0.010%	0.0002%	0.0196
	25920000	0.00001	0.001%	0.0001%	0.0621

Table 13 – CoV Calculations with Message Period 100ms

Appendix D Normalizing Measurements for PDR (Informative)

This document has specified a binning approach for delay-related measurements. When making measurements of delay variation, normalization is needed.

For the IPDV performance metric, a pair of delay values are normalized by subtracting one from the other, and taking the absolute value. Thus, the minimum of any IPDV measurement is 0, and as a consequence bins can be set up without any consideration for the actual magnitude of the delay.

A similar normalization is needed for PDR. PDR is defined as the difference between the Y^{th} percentile of delay and the minimum delay, so each delay observation needs to have the estimated minimum subtracted from it, to get a normalized delay. The PDR performance objective *O* is specified relative to a minimum of zero, as shown below in Figure 33.

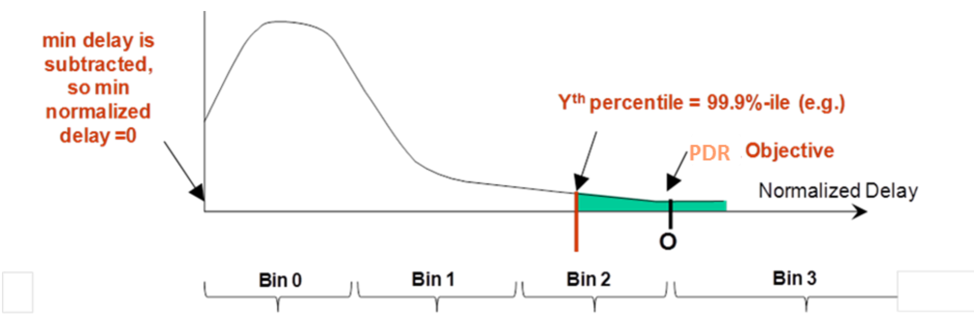

Figure 33 – Example PDR Distribution (normalized), and Bins

The distribution of delay is generally observed to be skewed to the right; i.e., there would be many measurements at or near the minimum delay, and fewer at higher values. Therefore, a good estimate of the minimum can be determined in a time interval much shorter than a Measurement Interval. Once an estimate of the minimum is available, observed delays can be normalized by subtracting the minimum, and then the appropriate bin counters can be incremented as the normalized delay is processed from each received IP SOAM Measurement packet.

One suggested practical approach as shown in Figure 33 is to record the minimum delay of each Measurement Interval, and to use that value as the estimated minimum at the beginning of the following Measurement Interval. As each delay measurement is received, the estimated minimum can be set to the minimum of the current measured delay and the previous estimate. Then each received delay measurement is normalized by subtracting the estimated minimum. With this approach, there would never be a negative value for a normalized PDR measurement.

Very small shifts in the minimum could be observed that would not be significant. Define ε as the threshold below which a shift is not considered significant (e.g., 10% of the objective). Then the SOF/ICM would not take actions if the shift of the minimum was less than ε. If, on the other hand, the minimum at the end of a Measurement Interval has decreased / increased by a value more than ε, the SOF/ICM is expected to consider as invalid the PDR measurements in the associated Measurement Interval(s).

If there are network changes during the Measurement Interval, then PDR measurements during that Measurement Interval may be invalid, and the measurements can be ignored by the SOF/ICM. This is discussed next. However, other MIs would still be valid and contribute to the estimate of PDR during the interval *T*.

Note that this approach is presented as an example, and that alternate implementations may improve on it.

D.1 Topology Shifts

For a fixed topology, the minimum delay is essentially fixed. However, network changes (e.g., in response to a network failure) can result in a shift in the minimum delay that can be significant. The minimum delay can of course shift to a lower or to a higher value.

D.1.1 Minimum Delay Becomes Significantly Smaller

When the delay becomes significantly smaller, as is shown in MI 2 below in Figure 34, it will be obvious at the end of MI 2 that the minimum delay is significantly lower than the minimum delay at the end of MI 1. It would be straightforward for an SOF/ICM to simply consider the PDR measurements of that interval as being invalid, and to ignore them.

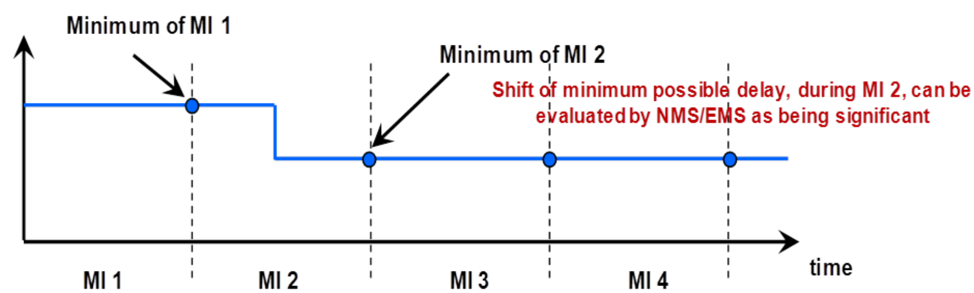

Figure 34 – Reduction in Minimum Delay, due to Network Topology Change

D.1.2 Minimum Delay Becomes Significantly Larger

When the delay becomes significantly larger, as is shown in MI 6 below in Figure 35, it will not be obvious until the end of MI 7 that the minimum delay is significantly higher than the minimum delay observed at the end of MI 5. It would be straightforward for the SOF/ICM to detect that and mark the measurements of MI 6 and MI 7 as being invalid.

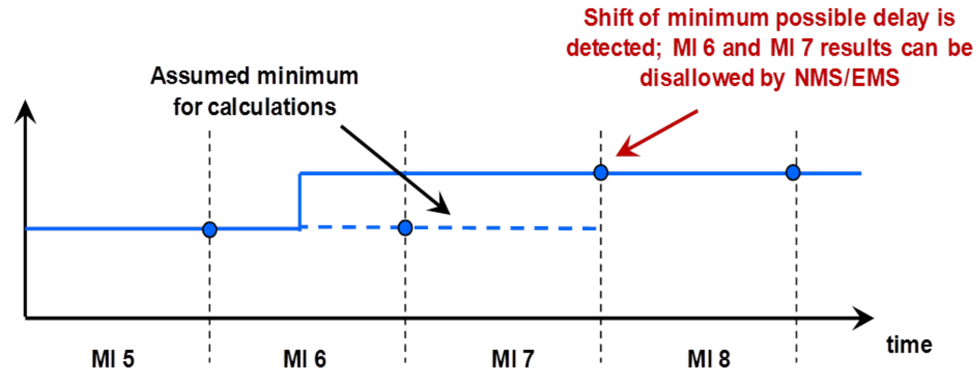

Figure 35 – Increase in Minimum Delay, due to Network Topology Change

D.2 Impact of Lack of ToD Synchronization

When performing One-way measurements using Single-Ended Delay Measurement without ToD synchronization between the MPs, negative packet delay measurements can be seen due to differences in the ToD for each MP. An example of this is shown in Figure 36.

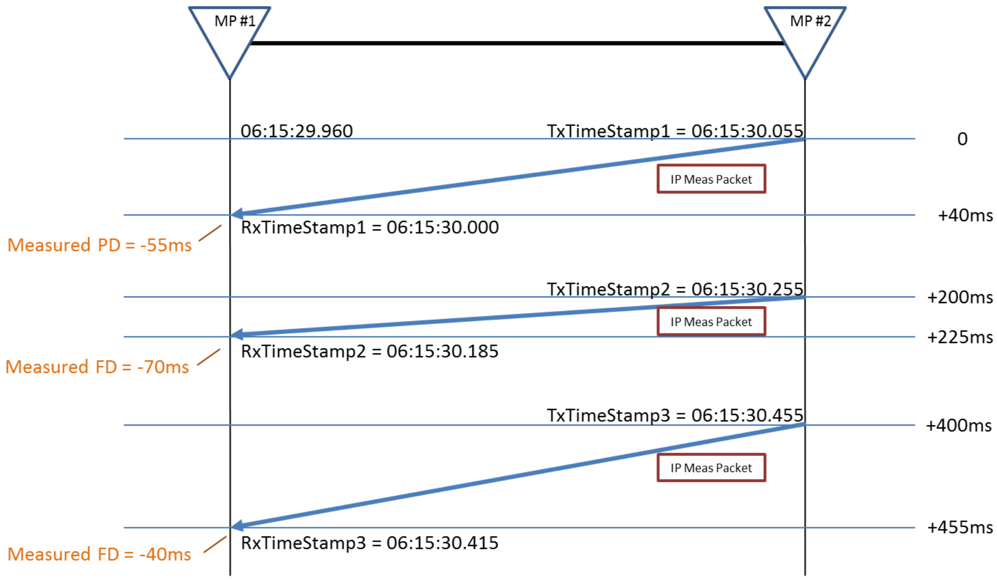

Figure 36 – Lack of ToD Synchronization

In Figure, three IP SOAM Measurement Packets are shown. At the time when the first measurement packet is transmitted, the ToD clock at MP #1 reads 06:15:30.055 and the ToD clock at MP #2 reads 06:15:29.960. The PD measured for the first packet, using RxTimeStamp1 – TxtimeStamp1, is -55ms since TxTimeStamp1 > RxTimeStamp1. When determining the minimum PD for PDR in this situation, a “less negative” PD is considered an increase in delay and a “more negative” PD is considered a decrease in delay. Using the example in Figure, the PD measured for the second packet, RxTimeStamp2 – TxTimeStamp2, is -70ms which indicates that the packet arrived 15ms faster than the first packet. The PD measured for the third packet, RxTimeStamp3 – TxTimeStamp3, is -40ms which indicates that the packet arrive 15ms slower than the first packet.

Implementations that are measuring PDR without ToD synchronization are expected to take this into account and react accordingly to negative PD measurements.

Appendix E Calculation of SLS Performance Metrics (Informative)

This document defines the data sets that devices or virtual applications provide to SOF/ICM, while other MEF specifications and applications need to obtain the performance metrics for SLS. This appendix provides some guidelines for how to calculate SLS performance metrics, using data sets as inputs.

The SLS performance metrics are defined in terms of the performance of every Qualified Service Packet; however, the data sets are primarily based on time-based samples. In the remainder of this appendix we assume that time-based sampling is used, and analyze how the data sets can be used to calculate the SLS metrics on that basis.

The data sets are Measurement Interval based. Traditionally, the duration of a Measurement Interval is 15 minutes or 24 hours. This document requires at least that 15 minute Measurement Intervals are supported. When reaching the end of a Measurement Interval, the data set for the current measurement interval is moved to the list of historic Measurement Intervals. The SOF/ICM can retrieve a block of historic data sets from the devices or virtual applications or they are transmitted to the SOF/ICM. Usually the performance metrics are measured against the SLS over a much longer time period T, typically one month or so. The processing of performance metrics for an SLS can be done by ICM, SOF or even the Business Systems. Therefore, the data sets from multiple Measurement Intervals are used for calculating the performance metrics over period T. In the following, we discuss how to obtain the following performance metrics for SLS, using IP SOAM PM defined data sets:

- One-way PD
- One-way MPD
- One-way PL

E.1 One-way Packet Delay

The one-way packet delay for an IP Data Packet that flows between SLS-RP i and SLS-RP j is defined as the time elapsed from the reception of the first bit of the packet at SLS-RP i until the transmission of the last bit of the first corresponding egress packet at SLS-RP j. If the packet is erroneously duplicated as it traverses the network, the delay is based on the first copy that is delivered.

One-way PD can be calculated from the data sets (i.e. counts of each Measurement Bin), when there are n Measurement Intervals in T for each CoS Name (*C*), and each set of ordered pair of SLS-RPs (*S*) in the SLS.

If PD(T) (%) <= $\hat{d}$ the SLS performance objective, then the performance is considered to meet the SLS for time period T. The PD over T can be calculated from:

$$PD(T) = \frac{\sum^{n} (Total\ counts\ of\ Meas. Bins\ in\ the\ MI\ that\ meet\ the\ objective}{\sum^{n} (Total\ counts\ of\ all\ Meas. Bins\ in\ the\ MI}$$

Note that the Measurement Bin thresholds must be chosen such that the PD objective $\hat{d}$ is aligned with the boundary between two bins, as described in Appendix B.

The same calculation applies to all other SLS performance metrics for which Measurement Bins are used, including One-way PDR and One-way IPDV.

E.2 One-way Mean Packet Delay

One-way Mean Packet Delay is defined in MEF 61.1 as:

- Let $\mu(T_k, C, <i, j>)$ represent the arithmetic mean of one-way packet delay for all Qualified Packets for time period T_k, CoS Name C and pair of MPs of SLS-RPs $<i, j>$ in S that are delivered to SLS-RP j. If there are no such packets, let $\mu(T_k, C, <i, j>)$ equal 0.

- Then the One-way Mean Packet Delay Performance Metric $u(T_k, C, S)$ is the maximum of the values $\mu(T_k, C, <i, j>)$ for all $<i, j>$ in S.

Since the MPD is calculated based on data sets for each CoS Name (C), and each set of ordered pair of SLS-RPs (S) in the SLS, where there are n MIs in T is:

$$MPD(T) = \frac{\sum^{n} (MPD\ of\ MI)}{n}$$

Where $\hat{u}$ is the objective for MPD.

MEF 35.1 Appendix I discusses other possible methods but agrees that this is the preferred method. See MEF 35.1 for information on the other methods.

E.3 One-way Packet Loss

MEF 61.1 [33] defines One-way Packet Loss Ratio as:

- Let $I(T_k, C, <i, j>)$ be the number of Qualified Packets for time period T_k, CoS Name C and ordered pair of SLS-RPs $<i, j>$ in S that are received at SLS-RP i.
- Let $J(T_k, C, <i, j>)$ be the number of unique (not duplicate) Qualified Packets for time period T_k, CoS Name C and ordered pair of SLS-RPs $<i, j>$ in S that are transmitted at SLS-RP j.
- Let $f(T_k, C, <i, j>)$ be defined as:
  $f(T_k, C, <i, j>) = \frac{I(T_k,C,<i,j>) - J(T_k,C,<i,j>)}{I(T_k,C,<i,j>)}$ if $I(T_k, C, <i, j>) > 0$

Based on the Tx and Rx packet counts of the data sets for n MIs during T, the One-way Packet Loss Ratio over T can be obtained by:

$$PLR(T) = \frac{\sum^{n}((Tx\ packet\ counts\ for\ the\ MI) - (Rx\ packet\ counts\ for\ the\ MI))}{\sum^{n}(Tx\ packet\ counts\ for\ the\ MI)}$$

Where $\hat{F}$ is the objective for the Packet Loss Ratio SLS.