

Internet Draft  
SFC Working Group  
Category: Informational  
Expires: December 30, 2016

Anil Kumar S N  
Gaurav Agrawal  
Vinod Kumar S  
Huawei Technologies  
C. Jacquenet  
Orange  
June 28, 2016

Performance Measurement Architecture for SFC  
draft-agv-sfc-performance-measurement-architecture-02

Abstract

This document describes performance measurement(PM) architecture for Service Function Chains (SFCs) to assess the operational status and behavior of a service function, a subset of service function's, a whole service function chain as a function of the actual deterministic service function / service function chain. This document does not propose solutions, protocols, nor any extension to existing protocols.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2016

## Copyright and License Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Architecture Overview . . . . .	7
3 PM Measurements Methods: . . . . .	8
4 Performance Measurement Attributes . . . . .	9
4.1 Metadata Attributes . . . . .	9
4.2 PM Reporting Attributes . . . . .	9
4.3 Performance Measurement Attributes Description . . . . .	9
4.3.1 PMF Identifier . . . . .	10
4.3.2 Window Identifier . . . . .	10
4.3.3 Measurement Agents Identifier with PM type . . . . .	11
4.3.4 Performance Statistics . . . . .	12
5 Measurement Controller Operation . . . . .	12
6 Measurement Classifier Operation . . . . .	13
7 MA Operation . . . . .	13
8 Collector Operation . . . . .	14
9 Measurement Steps . . . . .	14
10 Hop by Hop Performance Measurement . . . . .	15
11 SFC as a whole Performance Measurement . . . . .	15
12 Acknowledgements . . . . .	15
13 Security Considerations . . . . .	16
14 IANA Considerations . . . . .	16
15 References . . . . .	17
15.1 Normative References . . . . .	17
Authors' Addresses . . . . .	19



## 1. Introduction

The delivery of end-to-end services often requires various service functions. These include traditional network service functions such as firewalls and traditional IP network address translators (NATs), as well as application-specific functions. The definition and instantiation of an ordered set of service functions and subsequent "steering" of traffic through them is termed service function chaining (SFC).

This document describes an architecture used for assessing and monitoring SFC operation. It includes architectural concepts, principles, and components, with a focus on assessing the operational status and behavior of a SF, a subset of SFs, a whole SFC as a function of the actual deterministic SF/SFC.

Proper operation of a SFC depends on the ability to monitor and quickly identify faults and focus attention on the root cause of the problem. To achieve this we need to monitor the performance parameters of SFC like packet loss, delay, jitter etc.

For instance a SFC is composed of diffServ (e.g., traffic conditioning capabilities) and NAT functions, with OOP as a traffic type. To ensure the proper operation of a OOP it's required to ensure that traffic is properly re-marked before invoking the NAT function which selects a specific pool for such OOP traffic. Incorrect re-marking of this traffic will lead to performance deterioration in terms of increase in one way delay.

For another instance a SFC is composed of firewall which is supposed to controls the incoming and outgoing network traffic based on predetermined security rules. To ensure the proper functioning of this SF it's required to ensure that a specific set of traffic must be dropped and another specific set of traffic must be allowed to pass.

Continuous measurement and monitoring of packet delay/loss for a SF, a subset of SFs, a whole SFC will help to find the problem. Also performance measurement results can be used to locate the root cause. which inturn can also be used to possibly take required action as far as SFC structuring is concerned.

Service provider's service level agreements (SLAs) usually include performance indicators like packet loss or inter-packet delay variation that need to be measured over a given period of time. SFC Performance measurement enables operators with greater visibility into the performance characteristics of their SFCs, thereby ensuring service SLAs.

SFC is at service layer which is independent of transport layer technology. For a SFC underlying layer can be a mix of multiple transport layer tunneling technology like MPLS, VXLAN etc. SLA measurements of independent transport layer will not be capable to assess the operational status and behavior of a SF, a subset of SFs, a whole SFC as a function of the actual deterministic SF/SFC which leads to a requirement for performance measurement architecture specific to service function chain.

Transport layer PM and SFC PM can go hand in hand, SFC PM can be used to obtain the exact segment of problem if that segment is one of the Transport layer Tunnel, transport layer PM can be used to locate exact fault location.

Requirements of SFC performance measurement architecture includes:

- 1) Localization of performance degradation occurred due to a unit in a service function chain.
- 2) Localization of performance degradation for a specific service traffic type
- 3) Enable service providers with a tool to offer high value services whose performance degradation can be proactively located and corrected.
- 4) ECMP scenarios can result in out of order packets, performance measurement should consider this while measurement.
- 5) Validation of service function chain performance at the time of deployment/fault correction verification (using probe packets)
- 6) Runtime Validation of service function chain performance using traffic packets itself to get accurate performance data without additional probe packets.
- 7) Performance measurement under induced traffic level. The motivation for this can be related to the rate and the amount of traffic that can influence the accuracy of the measurement results
- 8) Optimized measurement by keeping minimal performance measurement load.

Following capabilities are required to be supported by performance measurement architecture to address above requirements.

- 1) Capability to assess and monitor at
  - a) SF
  - b) SFF
  - c) Set of SF/SFF
  - d) Segment(s) between any two SF/SFF in a SFC.
  - e) SFC as a whole.
- 2) Capability to measure performance for fine-grained and coarse-grained flow.
- 3) Capability for Continuous/proactive & selective/on-demand measurement.
- 4) Support for all three measurement methods:

Active Measurement method:

Active method measures performance and other parameters by injecting test traffic into the network from specific locations (e.g., a Point of Presence).

Passive measurement method:

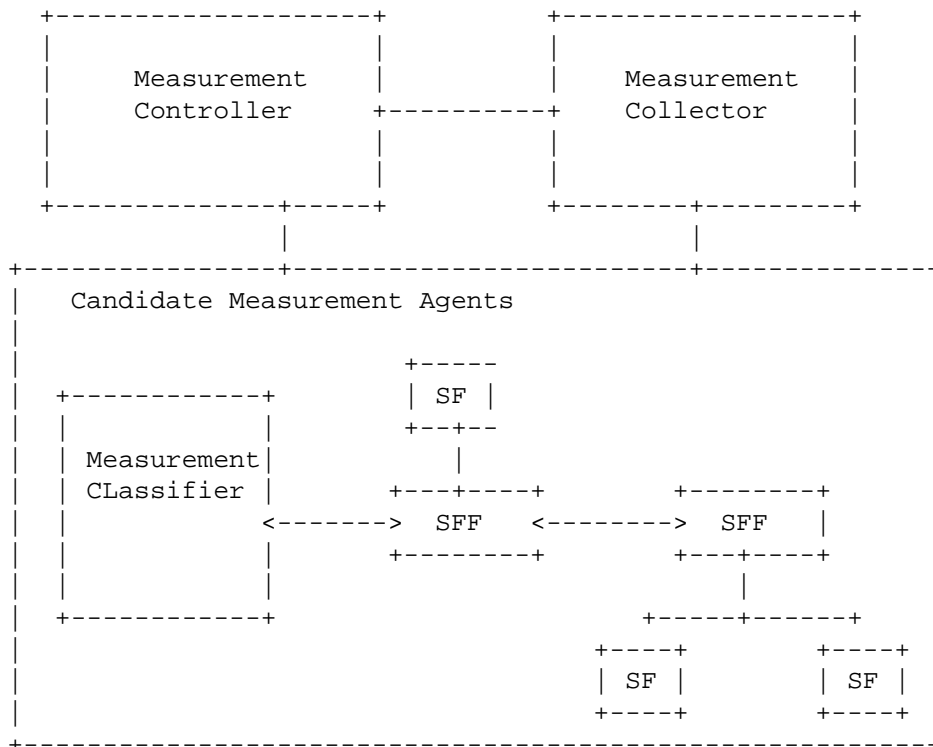
Passive method measures performance and other parameters based on the observation of live traffic forwarded in the network.

Hybrid measurement method:

Hybrid measurement method uses both active and passive measurement methods.

- 5) Capability to measure performance even in case of out of order packets.

## 2. Architecture Overview



The SFC performance measurement architecture relies upon the following components:

**Measurement Controller:** An entity that controls and coordinates a set of measurement functions. The measurement controller is responsible for programming the performance measurement instance at measurement classifier and updating the same to measurement collector, additionally it can inject probe packets in SFP for measurement.

**Measurement Collector:** An entity that collects measurement data from a measurement agents (MAs). A measurement collector functionality could be integrated with one of the MAs deployed in the network or with the measurement controller itself.

**Measurement Classifier:** An entity responsible for encoding measurement control information based on PMI programmed by measurement controller. SFC Classifier MUST incorporate the measurement classifier functionality to support SFC performance measurement.

Measurement Agent: An entity that supports one or more Measurement functions. It is responsible for performance data collection and reporting the same to Measurement collector.

The following elements MUST incorporate the MA functionality to support SFC performance measurement.

- o SF
- o SFF
- o Classifier
- o NSH Proxy Agent.

### 3 PM Measurements Methods:

- a) Active measurement : Measurement controller induce probe packets with encoded performance measurement data or programs PMI at measurement classifier which will in turn induce probe packets with encoded performance measurement data. measurement controller updates the PMI to measurement collector. Participating measurement agents based on received encoded information carry out measurements and reports the collected data to measurement collector. measurement collector co-relates received data and provides measurement results.

Note: Current draft only covers the Measurement attributes programming for probe packets. Construction of base probe packet is outside the scope of this RFC.

- b) Passive measurement : Measurement controller programming PMI at measurement classifier and updating the same to measurement collector. measurement classifier encodes the performance measurement data in classified packets. Participating measurement agents based on received encoded information carry out measurement and reports the collected data to measurement collector. measurement collector co- relates received data and provides measurement results.
- c) Hybrid measurement : Measurement controller induce probe packets to pre-setup load condition for a SFC using active measurement. measurement controller than uses passive measurement to carry out performance measurement under load condition.



## 4 Performance Measurement Attributes

There are two types of PM attributes

- a) Metadata Attributes: These attributes are to instruct the MA to carry out the required performance measurement. Measurement classifier encodes these attributes in the traffic packets based on instructions provided by controller or these attributes are encoded directly in test packet by measurement controller/measurement classifier.
- b) PM reporting attributes: These attributes are used by MA(s) to report the collected performance data to Collector

### 4.1 Metadata Attributes

The following attributes must be encoded as metadata for selected traffic or probe packets

- PMF Identifier
- Window Identifier
- List of MA(s) identifier with PM Type

### 4.2 PM Reporting Attributes

The following Information should be sent by each MA to the collector:

- PMF identifier
- Window identifier
- MA identifier with PM type
- Performance statistics

### 4.3 Performance Measurement Attributes Description

Several performance attributes are introduced in this memo to carry performance control information. Each attribute has a unique purpose; they are interpreted based upon a specific context and the corresponding performance measurement is performed by the respective MA.

#### 4.3.1 PMF Identifier

Purpose : To unambiguously identify a measurement flow in the SFC Domain. When a MA reports its performance data to collector, collector has to associate the data to a particular instance of measurement, as controller would have created multiple instance of measurements. By using service path index + service index only coarse-grained flows can be identified. To identify a fine-grained flows (Ex: SFC classification is based on destination IP, so flow classified based on this destination IP is coarse-grained flow. Sub flow corresponding to a particular destination IP within this flow is an example of fine-grained flows) within a SFC domain SPI + SI is not enough. PMF identifier is used to uniquely identify the performance measurement instance corresponding to fine grained flows.

- Value : Unique value within current SFC domain
- Processing :
- At Classifier : The PMF Identifier is encoded in metadata  
Note: For probe packets this information is encoded by entity (controller/classifier) constructing probe packet.
  - At MA : Used as a key while collecting, maintaining & reporting PM statistics to collector.
  - At Collector : Collector correlates the performance data received from a MA using this PMF Identifier.

#### 4.3.2 Window Identifier

Purpose : Window identifier is to handle out of order packet scenario.

Example:

```
SFC classifier-----SF1-----SF2-----SF3
Packet 1-->----->Packet 2
Packet 2-->----->Packet 1
```

Packet 1 belongs to earlier cycle of measurement but SF3 receives Packet 2 before packet 1 due to out of order problem. Now, when this data is co-related at Collector, if there is no window identifier, packet 2 will be treated as packet 1 and result will be incorrect. Window identifier is of local scope to a MA.

Sender (Classifier/Controller) divides the flows into multiple windows. Packets with PM information in a window will have the

same window identifier and consecutive windows will have a different identifier. This enables MA to collect & accumulate statistics corresponding to each window & report it to collector. Size of the window is programmable.

Value : Integer (Max/Min Value: Programmable to context header at classifier, once Max value reached then value = Min Value). Value increments with each PM interval.

Processing :

- At Classifier : The window identifier is encoded in NSH context header.
- At MA : Used as a key while collecting, maintaining & reporting PM statistics to collector.
- At Collector : Collector correlates the performance data received from MA by using this window identifier.

#### 4.3.3 Measurement Agents Identifier with PM type

Purpose : To identify participating measurement agents with context and type of performance measurement.

Value : Measurement agent for a given SFP is identified using MA identifier + service index

MA identifier has two parts

- 1) Node identifier - 24 bit
  - a) For SFF: MUST be unique number assigned by controller
  - b) For SF: All zero. Context identifier itself identifies SF node.
- 2) Order identifier - 8 bit
  - a) For SFF: Service index of next SF.
  - b) For SF: Service index

MA identifier SHOULD be in decreasing order of the SI for optimized traversal of the SI participation.

PM Type (IANA assigned values)

Processing :

- At Classifier : Encode the participating MA(s) with PM type.

- At MA : Presence of self service index triggers the PM collection & reporting
- At Collector : Identifies the reporting MA

#### 4.3.4 Performance Statistics

- |                 |   |
|-----------------|---|
| Purpose         | : Computation of performance.             |
| Value           | : Collected statistics                    |
| Processing      | :   |
| - At Classifier | : None.                                   |
| - At MA         | : Depends on performance measurement type |
- For packet loss: Accumulates received & sent packets counter for a given flow + window and reports it to collector
- For packet delay: Record the time for sending and receiving a packet of a given flow + window and reports it to collector.
- At Collector : Correlates and maintains received data

### 5 Measurement Controller Operation

The Measurement Controller has the following responsibilities:

- 1) Programs the unique MA identifier for SF/SFF/SFC proxy in SFC domain.
- 2) Programs the PM instance at the classifier
  - PM Instance MUST contain the following information:
    - a) PM flow classification rule with PMF identifier (unique across the SFC domain). For coarse grained flow PM flow classification rule is optional.
    - b) List of participating MA with PM type.
    - c) Window size and PM schedule.
- 3) Provides the PM Instance to the collector (if required)
- 4) Ensures the uniqueness of MA identifier & PMF identifier across the SFC domain
- 5) Programs the reporting interval at the MAs (optional).
- 6) For active measurement creates and send test probe packet.

## 6 Measurement Classifier Operation

The classifier classifies packets for the PM instance based on the instructions provided by the controller. In the classified packets it encodes the following information in context header of NSH metadata:

PMF identifier	: As programmed by controller
Window identifier	: Locally generated number which changes based on the window size.
MA(s)	: List of MA(s) with PM type

For active measurement measurement classifier MAY need to generate probe packet as instructed by measurement controller.

Note: Selection of packets in a flow to participate in a PM is decided by controller and it is outside the scope of this document.

## 7 MA Operation

MA carries out the following operations upon receipt of a packet with NSH header:

- Detection of PM context header in a packet.
- Processing of context header information.
- Check the presence of self index in context header.
- Identification of the PM type.
- Performance measurement based on the identified type.
  - \* For packet loss: Accumulates statistics for received & sent packets for a given PMF + window
  - \* For packet delay: Record the time when the packet was received and sent for a given PMF + window
- Reporting of accumulated statistics at configured interval to the collector. This interval should be consistent across all MAs.

Note:

- 1) A MA does not maintain the context of the window, the statistical information of a single window can be sent in more than one report, it is the collector's responsibility to map and accumulate statistics of a window from different reports.
- 2) If the classifier itself embeds a MA, it also needs to do performance measurement and reporting.

## 8 Collector Operation

Collector collects data from the MA and performs the following operations for data correlation purposes:

- Collector uses PMF identifier to group statistics related to a given PM flow. Collector maintains the statistics related to a given PM flow from multiple MAs to assess the forwarding performance.
- Collector uses the window identifier to group statistics received from multiple MAs for a single window for a given PM flow.
- Since the MA does not maintain the context of the block interval, statistical information of a single block can be received in more than one report; collector maps and accumulates the statistics of a block from different reports.
- Collector performs the PM computation based on the PM type & statistics collected from the MA; the identification of PM segment is done in the collector, based on the PM types received from the segment end point MAs.

## 9 Measurement Steps

Step 1: Programming of MA identifier at SF/SFF/SFC proxy by controller. Programming of PM instance at classifier by controller

Step 2: Classifier classifies & select packets for a PM Flow.

Step 3: Classifier encapsulates the PM attributes in PM context header for selected packets.

Step 4: Packet is sent out of classifier.

Step 5: MA receives packet and check presence of PM context header (If context header is not present move to step 9.)

Step 6: MA checks presence of its service index in PM context header (If its own index is not present, then move to Step 9)

Step 7: MA obtains PM type to be carried out and according accumulates PM statistics for a given PMF + window for received and sent packets.

Step 8: MA reports the accumulated PM statistics to collector at reporting intervals.

Step 9: Regular packet processing continues.

Step 5 to step 9 is repeated at every MA.

#### 10 Hop by Hop Performance Measurement

In case PM needs to be performed on all the SFs invoked along the SFP, as per the described NSH programming in the classifier, all the SFs MA identifier needs to be added in the context header. This will ensure all the SFs/SFFs participate in the PM. This mechanism will require all the SFs/SFFs to traverse the context header until the self SI is found.

Alternatively, we can optimize this process by defining a new context type where all the SFs need to perform the specified PM. Thus, the classifier can skip the MA identifier encoding in the context header, and the MA can skip the MA identifier processing.

Extension for SFF participation in hop-by-hop PM. As mentioned in the previous section, we can define a special context types which means SFF and/or needs participate in the PM.

#### 11 SFC as a whole Performance Measurement

In case PM needs to be performed on the endpoint SFs along the SFP, these 2 MA identifier needs to be encoded in the context header, and will follow the procedure to perform E2E SF's PM.

In case the PM needs to be performed between the classifier and SFC domain boundary SFF, a special context type will be encoded in the NSH, which needs to be processed by the boundary SFF to report the PM data to the collector and then strip the NSH.

#### 12 Acknowledgements

Special Thanks to Nobin Mathew for his review and comments.

### 13 Security Considerations

There are no security considerations relevant to this document.

### 14 IANA Considerations

No actions are required from IANA as result of the publication of this document.



## 15 References

### 15.1 Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", [RFC 2330](#), May 1998.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", [RFC 2678](#), September 1999.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", [RFC 2679](#), September 1999.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", [RFC 2680](#), September 1999.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", [RFC 3148](#), July 2001.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", [RFC 3393](#), November 2002.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", [RFC 3432](#), November 2002.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", [RFC 4656](#), September 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", [RFC 7498](#), DOI 10.17487/

[RFC7498](#), April 2015,  
<<http://www.rfc-editor.org/info/rfc7498>>.

[SFC-arch]

Quinn, P., Ed. and J. Halpern, Ed., "Service Function  
Chaining (SFC) Architecture", 2014,  
<<http://datatracker.ietf.org/doc/draft-quinn-sfc-arch>>.

## Authors' Addresses

Anil Kumar S N  
Huawei Technologies India Pvt. Ltd,  
Near EPIP Industrial Area,  
Kundalahalli Village,  
Whitefield,  
Bangalore - 560066

EMail: anil.sn@huawei.com

Gaurav Agrawal  
Huawei Technologies India Pvt. Ltd,  
Near EPIP Industrial Area,  
Kundalahalli Village,  
Whitefield,  
Bangalore - 560066

EMail: gaurav.agrawal@huawei.com

Vinod Kumar S  
Huawei Technologies India Pvt. Ltd,  
Near EPIP Industrial Area,  
Kundalahalli Village,  
Whitefield,  
Bangalore - 560066

EMail: vinods.kumar@huawei.com

Christian Jacquenet  
Orange  
Rennes 35000  
France

Email: christian.jacquenet@orange.com