Abstract

This document describes passive performance measurement (PM) architecture for Service Function Chains (SFCs) in a network. It includes architectural concepts and principles for composite services performance measurement when deployed as SFCs. This document does not propose solutions, protocols, or extensions to existing protocols.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html
Copyright and License Notice

Copyright (c) 2015 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1</td>
<td>Terminology</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Terms &amp; Definition</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Architecture overview</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Measurement Controller, Measurement Collector &amp; MA.</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Performance Measurement Attributes</td>
<td>11</td>
</tr>
<tr>
<td>4.1</td>
<td>NSH Context Header Attributes</td>
<td>11</td>
</tr>
<tr>
<td>4.2</td>
<td>PM Reporting Attributes</td>
<td>11</td>
</tr>
<tr>
<td>4.3</td>
<td>Attributes Description</td>
<td>11</td>
</tr>
<tr>
<td>4.3.1</td>
<td>PMF Identifier</td>
<td>11</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Window Identifier</td>
<td>12</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Measurement Agents with PM Type</td>
<td>12</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Performance Statistics</td>
<td>13</td>
</tr>
<tr>
<td>4.3.5</td>
<td>PM Instance</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Measurement Controller Operation</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Classifier Operation</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>MA Operation</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Collector Operation</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Measurement steps</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>Hop by Hop Performance Measurement</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>End to End Performance Measurement</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>Security Considerations</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>IANA Considerations</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>References</td>
<td>17</td>
</tr>
<tr>
<td>14.1</td>
<td>Normative References</td>
<td>17</td>
</tr>
<tr>
<td>14.2</td>
<td>Informative References</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Authors’ Addresses</td>
<td>19</td>
</tr>
</tbody>
</table>
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).

The purpose of this memo is to define a general framework for systematic passive performance measurement for the Network Service Header (NSH) encapsulated packets or frames on service chains.

Service provider’s service level agreements (SLAs) depend on the ability to measure and monitor performance metrics mostly for following parameters:

a) Packet Loss
b) Packet delay

Performance measurement capability enables operators with greater visibility into the performance characteristics of their networks, thereby operators can carry out facilitating & planning, troubleshooting, and network performance evaluation.

Performance measurement methods could be broadly into two categories:

Active Measurement method:
Active method measures performance or reliability parameters by the examining injected special traffic into the network, especially for the purpose of measurement by intended measurement point.

Passive measurement method:
Passive method measures performance or reliability parameter based of observing existing live traffic (packets) on the network.
Both passive and active measurement methods have their strengths and should be regarded as complementary. There are scenarios where active measurements alone is not enough or applicable and passive measurements are desirable along with active measurement method, one such reasons could be the rate, numbers and interval between the injected active measurement packets may affect the accuracy of the results and also the injected test packets may not be guaranteed to always be in-band with the data traffic in the network due to Equal Cost Multi-Path (ECMP).

Below are some of the basic requirements for PM architecture for SFC

- Hop by Hop, E2E, & SFC segment(s) Measurement.
- Measurement for Granular Flows in SFP as well as SFP as a whole
- Continuous/proactive & selective/on-demand measurement.
- Packet Out of Ordering shouldn’t impact the PM calculation
- Compliance with NSH Standards
- PM applicability between different component in SPF path to exactly locate the problem area (this includes Measurement between SF-SF, SF-SFF, SFF-SFF etc.)

This document describes efficient and accurate passive performance measurement architecture for Service Function Chains (SFCs) in a service function domain with conformance to above listed basic requirements.
1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2 Terms & Definition

Network Service: An offering provided by an operator that is delivered using one or more service functions. This may also be referred to as a "composite service". The term "service" is used to denote a "network service" in the context of this document.

Note: Beyond this document, the term "service" is overloaded with varying definitions. For example, to some a service is an offering composed of several elements within the operator’s network, whereas for others a service, or more specifically a network service, is a discrete element such as a "firewall". Traditionally, such services (in the latter sense) host a set of service functions and have a network locator where the service is hosted.

Classification: Locally instantiated matching of traffic flows against policy for subsequent application of the required set of network service functions. The policy may be customer/network/service specific.

Classifier: An element that performs Classification.

Service Function Chain (SFC): A service function chain defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification. An example of an abstract service function is "a firewall". The implied order may not be a linear progression as the architecture allows for SFCs that copy to more than one branch, and also allows for cases where there is flexibility in the order in which service functions need to be applied. The term "service chain" is often used as shorthand for service function chain.

Service Function (SF): A function that is responsible for specific treatment of received packets. A Service Function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). As a logical component, a service function can be realized as a virtual element or be
embedded in a physical network element. One or more Service Functions can be embedded in the same network element. Multiple occurrences of the service function can exist in the same administrative domain.

One or more service functions can be involved in the delivery of added-value services. A non-exhaustive list of abstract service functions includes: firewalls, WAN and application acceleration, Deep Packet Inspection (DPI), Lawful Intercept (LI), server load balancing, NAT44 [RFC3022], NAT64 [RFC6146], NPTv6 [RFC6296], HOST_ID injection, HTTP Header Enrichment functions, and TCP optimizer.

An SF may be SFC encapsulation aware (that is, it receives and acts on information in the SFC encapsulation) or unaware (in which case, data forwarded to the SF does not contain the SFC encapsulation). This is often referred to as "SFC aware" and "SFC unaware", respectively.

Service Function Forwarder (SFF): A service function forwarder is responsible for forwarding traffic to one or more connected service functions according to information carried in the SFC encapsulation, as well as handling traffic coming back from the SF. Additionally, an SFF is responsible for delivering traffic to a classifier when needed and supported, transporting traffic to another SFF (in the same or different type of overlay), and terminating the Service Function Path (SFP).

Metadata: Provides the ability to exchange context information between classifiers and SFs, and among SFs.

Service Function Path (SFP): The service function path is a constrained specification of where packets assigned to a certain service function path must go. While it may be so constrained as to identify the exact locations, it can also be less specific. The SFP provides a level of indirection between the fully abstract notion of service chain as a sequence of abstract service functions to be delivered, and the fully specified notion of exactly which SFF/SFs the packet will visit when it actually traverses the network. By allowing the control components to specify this level of indirection, the operator may control the degree of SFF/SF selection authority that is delegated to the network.

SFC Encapsulation: The SFC encapsulation provides, at a minimum, SFP identification, and is used by the SFC-aware functions,
such as the SFF and SFC-aware SFs. The SFC encapsulation is not used for network packet forwarding. In addition to SFP identification, the SFC encapsulation carries metadata including data-plane context information.

Rendered Service Path (RSP): Within an SFP, packets themselves are of course transmitted from and to specific places in the network, visiting a specific sequence of SFFs and SFs. This sequence of actual visits by a packet to specific SFFs and SFs in the network is known as the Rendered Service Path (RSP). This definition is included here for use by later documents, such as when solutions may need to discuss the actual sequence of locations the packets visit.

SFC-Enabled Domain: A network or region of a network that implements SFC. An SFC-enabled domain is limited to a single network administrative domain.

SFC Proxy: Removes and inserts SFC encapsulation on behalf of an SFC-unaware service function. SFC proxies are logical elements.

Network Node/Element: Device that forwards packets or frames based on outer header information. In most cases is not aware of the presence of NSH.

Network Overlay: Logical network built on top of existing network (the underlay). Packets are encapsulated or tunneled to create the overlay network topology.

Network Service Header: Data plane header added to frames/packets. The header contains information required for service chaining, as well as metadata added and consumed by network nodes and service elements.

NSH Proxy: Acts as a gateway: removes and insert SH on behalf of a service function that is not NSH aware.

Service Classifier: Function that performs classification and imposes an NSH. Creates a service path. Non-initial (i.e. subsequent) classification can occur as needed and can alter, or create a new service path.

Measurement Collector: An operational function that collects measurement data from a Measurement Agent. Measurement collector is responsible for collecting the Performance measurement Data from Measurement Agent. Measurement collector functionality could be integrated with one of
the MA or with the Controller itself.

Measurement Agent: An operational function that contains one or more Measurement functions. Measurement Agents is responsible for understand and analyze Performance measurement control information encoded in NSH metadata and perform the performance data collecting and report the same to Measurement collector with key information to identify performance measurement instance along with data collected.

Measurement Controller: An operational function that controls running, scheduling, and general coordination of Measurement functions by instructing a Measurement Agent using NSH metadata. Measurement Controller is responsible for Configuring the Performance measurement Instance. Optionally Performance measurement instance can be configured manually at the Ingress in which case Controller is not required
Architecture overview

SFC performance measurement architecture will have the following major components:

Measurement Controller:
Responsible for Programming the PM instance at the SFC classifier. Optionally PM instance can be configured manually at the Ingress in which case controller is not required.

Measurement Collector:
Responsible for collecting the PM Data from MA(s).

SFC Classifier:
Responsible for programming/encoding measurement control information for MA(s) to collect the appropriate PM statistics information as NSH metadata based on traffic classification & Measurement controller instructions.

Measurement Agent:
Measurement Agents responsible for collecting the PM Data. From now on will be referred as MA.
The following elements in a SFP can act as a MA:
- SF
- SFF
- Classifier
- NSH Proxy Agent.

Note: Controller and Collector function can be hosted on a single device which depending on deployment.

### 3 Measurement Controller, Measurement Collector & MA.

As described the major components of a service function enabled network performance measurement platform are the Measurement Agents, the Measurement Controller(s) and the Measurement Collector(s).

The MAs are the elements actually performing the measurements. The MAs are controlled by exactly one Controller at a time and the Measurement Collectors gather the results generated by the MAs.

In a nutshell, the normal operation of a SFC performance measurement platform starts with the Controller instructing a set of one or more MAs to perform a set of one or more performance measurement tasks at a certain point in time. The MAs execute the instructions from a Controller, and once they have done so, they report the results of the measurements to one or more Measurement Collectors.

The overall detailed framework for a SFC performance measurement platform along with information model will be described in subsequent draft versions.
4 Performance Measurement Attributes

To perform an effective and accurate measurement; Encoding metadata and accurate communication between measurement controller and measurement agent is very important.

To achieve desired goal of efficiency and accuracy many performance control information need to be encoded as NSH Context Header Attributes in NSH metadata. Once measurement is done then these key NSH Context Header Attributes along with measurement data need to be reported to measurement controller.

4.1 NSH Context Header Attributes

Following attributes should encoded in NSH metadata in a Context Header
- PMF Identifier
- Window Identifier
- List of MA(s) with PM Type

4.2 PM Reporting Attributes

Following Information should be sent from each MA to COLLECTOR
- PMF Identifier
- Window Identifier
- MA with PM Type
- Performance Statistics

4.3 Attributes Description

Multiple new performance attributes are introduced in this memo to carry performance control information. Each attribute has unique purpose; are understood based on context and corresponding performance is performed by respective MA in the SFP.

4.3.1 PMF Identifier

Purpose    : For unique identification of a measured Flow in SFC Domain

Value      : Unique value within current SFC domain

Processing :

- At Classifier : The PMF Identifier in encoded in NSH Context Header.
- At MA         : Used as a key while collecting, maintaining & reporting PM Statistics to Collector.
- At Collector: Collector co-relates the performance data received from MA using this PMF Identifier.

### 4.3.2 Window Identifier

**Purpose**: Divides the flows into multiple Windows. Packets with PM information in a Window will have same Window identifier and consecutive Windows will have 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 co-relates the performance data received from MA using this Window Identifier.

### 4.3.3 Measurement Agents with PM Type

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

**Value**: Service Index to identify SF in SFP.

**Processing**:
- At Classifier: Encode the Participating MA(s) with PM Type.
- At MA: Presence of self 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 report it to Collector
  For Delay: Record the time for sent and received packet for a given Flow + window and report it to Collector.
- At Collector : Co-relate and maintain received data

4.3.5 PM Instance

PM Instance is a set of parameters consisting PMF ID, MA List along with its PM Type, Window size & PM Schedule (Specifies the time when the PM has to be performed). PM Instance uniquely identifies a PM flow in SFC domain. PM Instance needs to be programmed for a classified flow at the classifier either by Controller or by manually configuration. Values of these parameters will depend on the measurement scenario.

5 Measurement Controller Operation

Measurement Controller has the following responsibility:

1) Program the PM Instance at the Classifier
2) Provides the PM Instance details to the Collector (if required)
3) Ensures the uniqueness of PMF Id across the SF Domain
4) Program the reporting interval at the MAs (optional).
6 Classifier Operation

Classifier classifies packets for the PM Instance based on the instruction provided by the controller. In the classified packets it encodes the following information in Context header of NSH metadata

PMF Identifier: As provided by the Controller
Window Identifier: Locally generated Number which changes based on the Window size.
MA(s): List of MA(s) with PM Type

Note: Selecting the packets in a flow to participate in a PM is decided by Controller/Classifier and it is outside the scope of this document.

7 MA Operation

MA carries out following operation when packet with NSH header encountered:

- 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 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 MA.

Note:
1) MA does not maintain the context of the window, the statistics information of a single window can be sent in more than one report, its Collector’s responsibility to map and accumulates the statistics of a window from different reports.
2) If Classifier itself is a MA it also needs to do performance measurement and reporting.
8 Collector Operation

Collector collects the data from the MA and performs the following for Data co-relation:

- Collector uses PMF Identifier to group statistics related to single PM flow. Collector maintains the statistics related to a single PM flow from multiple MA to compute the performance for the flow.
- Collector uses the Window identifier to group the statistics received from multiple MA for a single window for a given PM Flow.
- Since MA does not maintain the context of the block interval, statistics 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 computations 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 PM Instance at Classifier.

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 check presence of its service index in PM Context Header (If its own index is not present than 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 packet.

Step 8: MA reports the accumulated PM statistics to Collector at reporting interval.

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 in the SFP, as per the described NSH programming in classifier, all the SF’s, SI needs to be added in the context header. This will ensure all the SFs participate in the PM. This mechanism will require all the SF to traverse the context header until the self SI is found.

Alternatively, we can optimize this process by defining a new context type which means the all the SF’s needs to perform the specified PM. By this way at classifier we can skip the SI encoding in the context header, and at MA we can skip the SI traversing.

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

11 End to End Performance Measurement

In case PM needs to be performed on the end point SFs in the SFP, these 2 SI needs to be encoded in the context header, and it will follow the normal procedure to perform E2E SF’s PM.

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

12 Security Considerations

No specific security considerations for this document

13 IANA Considerations

No specific IANA considerations for this document
14 References

14.1 Normative References


14.2 Informative References


Authors’ Addresses

Anil Kumar S N
Huawei Technologies India Pvt. Ltd,
Near EPIP Industrial Area,
Kundalahalli Village,
Whitefield,
Bangalore - 560037
EMail: anil.sn@huawei.com

Gaurav Agrawal
Huawei Technologies India Pvt. Ltd,
Near EPIP Industrial Area,
Kundalahalli Village,
Whitefield,
Bangalore - 560037
EMail: gaurav.agrawal@huawei.com

Vinod Kumar S
Huawei Technologies India Pvt. Ltd,
Near EPIP Industrial Area,
Kundalahalli Village,
Whitefield,
Bangalore - 560037
EMail: vinods.kumar@huawei.com