Network Working Group
Internet Draft
Intended status: Standards Track
Expires: February 2018

R. Browne
A. Chilikin
Intel
T. Mizrahi
Marvell

August 31, 2017

Network Service Header KPI Stamping
draft-browne-sfc-nsh-kpi-stamp-02.txt

Abstract

This draft describes a method of inserting Key Performance Indicators (KPIs) into Network Service Header (NSH) encapsulated packets or frames on service chains. This method may be used to monitor latency and QoS configuration to identify problems with virtual links (vlinks), Virtual Network Functions (VNFs) or Physical Network Functions (PNFs) on the Rendered Service Path (RSP).

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/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on February 28, 2018.

Copyright Notice

Table of Contents

1. Introduction

Network Service Header (NSH), as defined by [NSH], defines a method to insert a service-aware header in between payload and transport headers. This allows a great deal of flexibility and programmability in the forwarding plane allowing user flows to be programmed on-the-fly for the appropriate Service Functions (SFs).

Whilst NSH promises a compelling vista of operational agility for Service Providers, many service providers are concerned about losing service and configuration visibility in the transition from physical appliance SFs to virtualized SFs running in the Network Function Virtualization (NFV) domain. This concern increases when we consider that many service providers wish to run their networks seamlessly in 'hybrid' mode, whereby they wish to mix physical and virtual SFs and run services seamlessly between the two domains.

This draft describes a generic method to monitor and debug service chains in terms of application latency and QoS configuration of the flows within a service chain. This method is compliant with hybrid architectures in which VNFs and PNFs are freely mixed in the service chain. This method also is flexible to monitor the performance and configuration of an entire chain or part thereof as desired. Please refer to [NSH] as background architecture for the method described in this document.

In particular, this draft proposes mechanisms to detect and debug performance issues based on timestamping flows within a chain and to detect and debug QoS configuration on the chain. The method described here is easily extensible to monitoring other KPIs also.

The method described in this draft is not an OAM protocol like [Y.1731] or [Y.1564] for example. As such it does not define new OAM packet types or operation. Rather it monitors the service chain performance and configuration for subscriber payloads and indicates subscriber QoE rather than out-of-band infrastructure metrics.

2. Terminology

2.1. Requirement Language

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 [RFC2119].

2.2. Definition of Terms

Classification: Locally instantiated policy and customer/network/service profile matching of traffic flows for identification of appropriate outbound forwarding actions.

First Stamping Node (FSN): Mark packets correctly. Must understand 5 tuple information in order to match Stamping Controller flow table.

Last Stamping Node (LSN): Reads all MD & export to system performance statistics agent or repository. Should also send NSH header - the Service Index (SI) will indicate if a PNF(s) was at the end of the chain. The LSN changes the SPI in order that the underlay routes the metadata back directly to the KPI database (KPIDB).

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 inserts 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.

Service Function (SF): A function that is responsible for specific treatment of received packets. A service function can act at the network layer or other OSI layers. A service function can be virtual instance or be embedded in a physical network element. One of multiple service functions can be embedded in the same network element. Multiple instances of the service function can be enabled in the same administrative domain.

Service Function Chain (SFC): A service function chain defines an ordered set of service functions that must be applied to packets and/or frames selected as a result of classification. The implied order may not be a linear progression as the architecture allows for nodes that copy to more than one branch. The term service chain is often used as shorthand for service function chain.

Service Function Path (SFP): The instantiation of a SFC in the network. Packets follow a service function path from a classifier through the requisite service functions.

Stamping Controller SC: The SC may be part of the service chaining application, SDN controller, NFVO or any MANO entity. For clarity we define the SC separately here as the central logic that decides what packets to stamp and how. The SC instructs the classifier on how to build the NSH header.

Stamp Control Plane (SCP): the control plane between the FSN and the SC.

Key Performance Indicator Database (KPIDB): external storage of Metadata for reporting, trend analysis etc.

2.3. Abbreviations

DEI	Drop Eligible Indicator
DSCP	Differentiated Services Code Point
FSN	First Stamping Node
KPI	Key Performance Indicator
KPIDB	Key Performance Indicator Database
LSN	Last Stamping Node
MD	Metadata
NFV	Network Function Virtualization
NFVI-PoP	NFV Infrastructure Point of Presence
NIC	Network Interface Card
NSH	Network Service Header
OAM	Operations, Administration, and Maintenance
PCP	Priority Code Point
PNF	Physical Network Function
PNFN	Physical Network Function Node
QoE	Quality of Experience

QoS	Quality of Service
QS	QoS Stamp
RSP	Rendered Service Path
SC	Stamping Controller
SCL	Service Classifier
SCP	Stamp Control Plane
SI	Service Index
SF	Service Function
SFC	Service Function Chain
SFN	Service Function Node
SFP	Service Function Path
SSI	Stamp Service Index
TC	Traffic Class
TS	Timestamp
VLAN	Virtual Local Area Network
VNF	Virtual Network Function
vSwitch	Virtual Switch

3. NSH KPI Stamping

A typical KPI stamping architecture is presented in Figure 1.

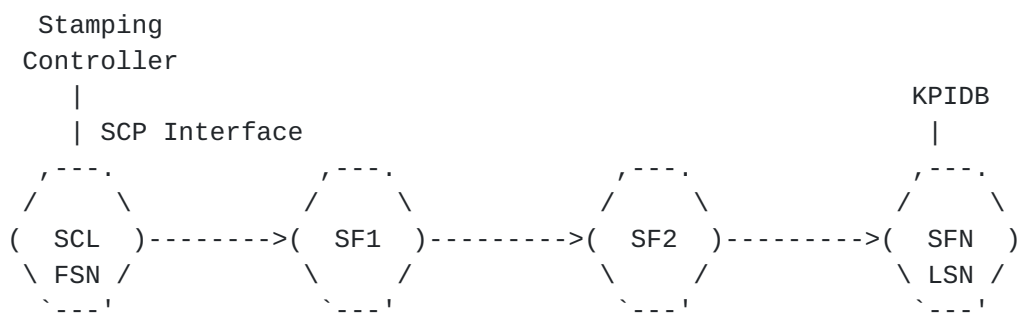

Figure 1: Logical roles in NSH KPI Stamping

The Stamping Controller (SC) will most probably be part of the SFC controller but is explained separately in this document for clarity. The SC is responsible for initiating start/stop stamp requests to the SCL or FSN, and also for distributing NSH stamping policy into the service chain via the Stamping Control Plane (SCP) interface.

The First Stamp Node (FSN) will typically be part of the SCL but again is called out as separate logical entity for clarity. The FSN is responsible for marking NSH MD fields for the correct flow with the appropriate NSH fields. This tells all upstream nodes how to behave in terms of stamping at VNF ingress, egress or both, or ignoring the stamp NSH metadata completely. The FSN also writes the Reference Time value, a (possibly inaccurate) estimate of the current time-of-day, into the header, allowing the {chain:flow} performance to be compared to previous samples for offline analysis. The FSN should return an error to the SC if not synchronized to the current time-of-day and forward the packet along the service-chain unchanged.

SF1, SF2 stamp the packets as dictated by the FSN and process the payload as per normal.

Note 1: The exact location of the stamp creation may not be in the VNF itself, as referenced in Section 3.3.

Note 2: Special cases exist where some of the SFs (PNFs or VNFs) are NSH-unaware. This is covered in Section 5.

The Last Stamp Node (LSN) should strip the entire header and forward the raw packet to the IP next hop. The LSN also exports NSH stamp information to the KPI Database (KPIDB) for offline analysis; the LSN may either export the stamping information of all packets, or a subset based on packet sampling. In fully virtualized environments the LSN will be co-located with the VNF that decrements the NSH

Service Index to zero. Corner cases exist whereby this is not the case and is covered in section 5.

3.1. Prerequisites

Timestamping presents a set of prerequisites not required to QoS-Stamp. In order to guarantee metadata accuracy, all servers hosting VNFs should be synchronized from a centralized stable clock. As it is assumed that PNFs do not timestamp there is no need for them to synchronize. There are two possible levels of synchronization:

Level A: Low accuracy time-of-day synchronization, based on NTP [RFC5905].

Level B: High accuracy synchronization (typically on the order of microseconds), based on [IEEE1588].

Each platform SHOULD have a level A synchronization, and MAY have a level B synchronization.

Level A requires each platform (including the Stamp Controller) to synchronize its system real-time-clock to an NTP server. This is used to mark the metadata in the chain, using the <Reference Time> field in the NSH KPIstamp header (Section 4.2). This timestamp is written to the NSH header by the first SF in the chain. NTP accuracy can vary by several milliseconds between locations. This is not an issue as the Reference Time is merely being used as a reference inserted into the KPIDB for performance monitoring.

Level B synchronization requires each platform to be synchronized to a Primary Reference Clock (PRC) using the Precision Time Protocol [IEEE1588]. A platform MAY also use Synchronous Ethernet ([G.8261], [G.8262], [G.8264]), allowing more accurate frequency synchronization.

If a SF is not synchronized at the moment of timestamping, it should indicate synch status in the NSH header. This is described in more detail in section 4.

By synchronizing the network in this way, the timestamping operation is independent of the current RSP, whether the entire chain is served by one NFVI-PoP or by multiple. Indeed the timestamp MD can indicate where a chain has been moved due to a resource starvation event as indicated in 0 below, between VNF 3 and VNF 4 at time B.

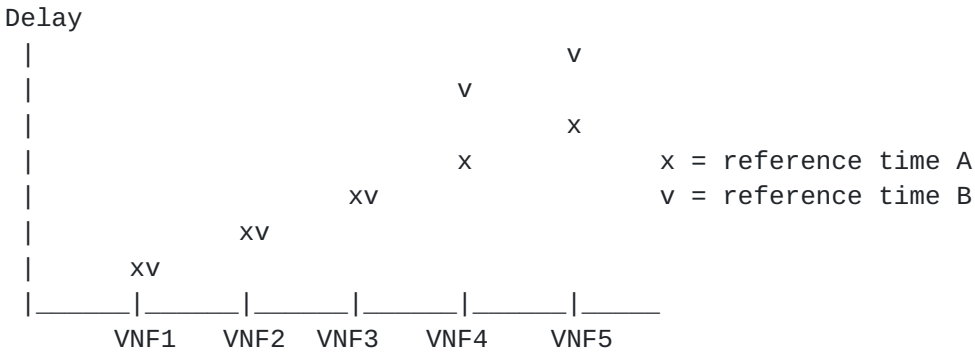

Figure 2: Flow performance in a service chain

For QoS Stamping it is desired that the SCL or FSN be synchronized in order to provide reference time for offline analysis, but this is not a hard requirement (they may be in holdover or free-run state for example). Subsequent upstream platforms do not need to be synchronized for QoS Stamping operation as described below

QoS stamping can be used to check consistency of configuration across the entire chain or part thereof. This will allow quick identification of QoS mismatches across multiple L2/L3 fields which otherwise is a manual, expert-led consuming process.

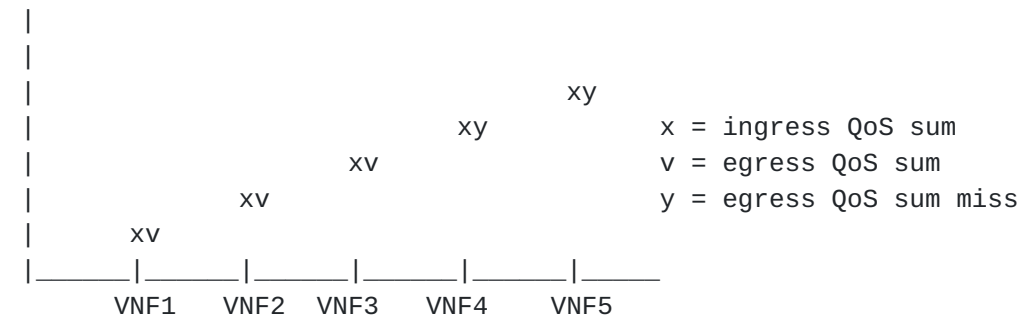

Figure 3: Flow QoS Consistency in a service chain

Referring to figure 3 above, x, v and y are notional sum values of the QoS configuration of the flow within a given chain. As the encapsulation of the flow can change from hop to hop in terms of VLAN header(s), MPLS labels, DSCP(s) these values are used to compare consistency of configuration from for example payload DSCP through overlay and underlay QoS settings in VLAN IEEE 802.1Q bits, TC MPLS bits and infrastructure DSCPs.

The above figure indicates that at VNF4 in the chain, the egress QoS marking is inconsistent. That is, the ingress QoS settings does not match the egress. The method described here will indicate which QoS field(s) is inconsistent, and whether this is ingress (whereby the underlay has incorrectly marked and queued the packet) or egress (where the VNF has incorrectly marked and queued the packet.

3.2. Operation

KPIstamping detection mode uses MD type 2. This involves the SFC classifier stamping the flow at chain ingress, and no subsequent stamps being applied, rather each VNF upstream can compare its local condition with the ingress value and take appropriate action. Therefore detection mode is very efficient in terms of header size that does not grow after the classification. This is further explained in section 4.1.

Section 3.5 of [NSH] (draft-ietf-sfc-nsh-10) defines NSH metadata type 2 encapsulation as per the figure below In KPIstamped detection and extended mode, flows will use this format.

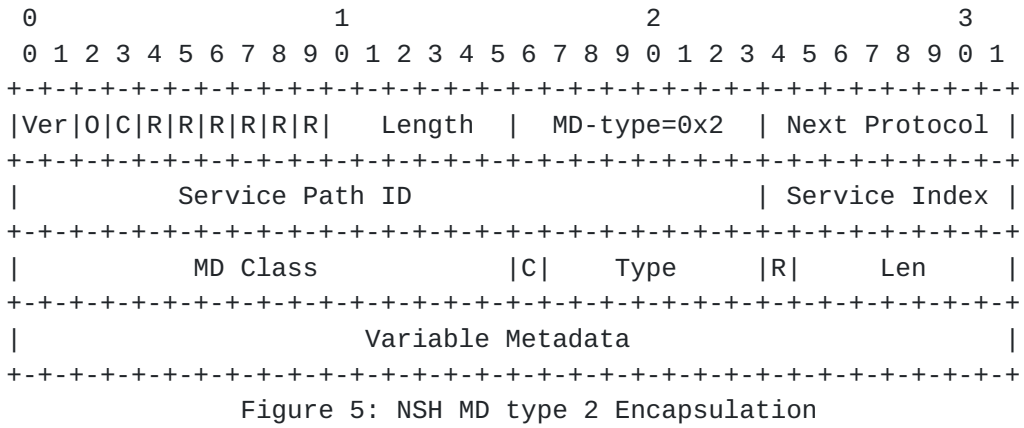

Figure 5: NSH MD type 2 Encapsulation

3.2.1. Flow Selection

The SC should maintain a list of flows within each service chain to be monitored. This flow table should be in the format SPI:5 tuple ID. The SC should map these pairs to unique Flow IDs per service chain within the extended NSH header specified in this draft. The SC should instruct the FSN to initiate timestamping on flow table match. The SC may also tell the classifier the duration of the timestamping operation, either by a number of packets in the flow or by a time duration.

In this way the system can monitor the performance of the all en-route traffic, or an individual subscriber in a chain, or just a specific application or QoS class the subscriber is running.

The SC should write the list of monitored flows into the KPIDB for correlation of performance and configuration data. Thus, when the KPIDB receives data from the LSN it understands to which flow the data pertains.

The association of source IP to subscriber identity is outside the scope of this draft and will vary by network application. For example, the method of association of a source IP to IMSI in mobile cores will be different to how a CPE with NAT function may be chained in an enterprise NFV application.

3.2.2. SCP Interface

A new Stamp control plane (SCP) interface is required between the SC and the FSN or classifier. This interface:

- Queries the SFC classifier for a list of active chains and flows
- Communicates which chains and flows to stamp. This can be a specific {chain:flow} combination or include wildcards for monitoring subscribers across multiple chains or multiple flows within one chain.
- How the stamp should be applied (ingress, egress, both or specific).

- Typically SCP timestamps flows for a certain duration for trend analysis, but only stamps one packet of each QoS class in a chain periodically (perhaps once per day or after a network change). Therefore timestamping is generally applied to a much larger set of packets than QoS stamping

- When to stop stamping, either after a certain number of packets or duration.

Exact specification of SCP is for further study.

3.3. Performance Considerations

This draft does not mandate a specific stamping implementation method, and thus NSH KPI stamping can either be performed by hardware mechanisms, or by software. If software-based stamping is used, applying and operating on the stamps themselves incur an additional small delay in the service chain. However, it can be assumed that these additional delays are all relative for the flow in question. This is only pertinent for timestamping mode, and not for QoS stamping mode. Thus, whist the absolute timestamps may not be fully accurate for normal non-timestamped traffic they can be assumed to be relative.

It is assumed that the method described in this document would only operate on a small percentage of user flows. The service provider may choose a flexible policy in the SC to timestamp a selection of user-plane every minute for example to highlight any performance issues. Alternatively, the LSN may selectively export a subset of the KPIstamps it receives, based on a predefined sampling method. Of course the SC can stress test an individual flow or chain should a deeper analysis be required. We can expect that this type of deep analysis has an impact on the performance of the chain itself whilst under investigation. The impact will be dependent on vendor implementation and outside the scope of this document.

For QoS stamping the method described here is even less intrusive, as you would not typically need to QoS stamp multiple packets in a flow rather periodically (perhaps once per day) check one packet in a chain per QoS class.

The KPIstamp may be applied at various parts of the NFV architecture. The VNF, hypervisor, vSwitch or NIC are all potential locations that can append the packet with the requested KPIstamp. Whilst it is desirable to stamp as close as possible to the VNF for accuracy, the exact location of the stamp application is outside the scope of this document, but should be consistent across the individual SC domain.

4. NSH KPIStamping Encapsulation

KPI stamping uses NSH MD type 0x02 for detection of anomalies and extended mode for root cause analysis of KPI violations. These are further explained in this section.

4.1. KPIstamping Encapsulation (Detection Mode)

The generic NSH MD type 2 allocation for KPI Stamping (detection mode) is shown below. This is the format we propose for KPI anomaly detection.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|O|C|R|R|R|R|R|R|   Length  |  MD type=0x2  | Next Protocol |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Service Path Identifier              | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    MD  Class=KPI Monitoring   |C| Type=TSD    |R|     Len       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|C|     KPIType  |      SI       |           Flow ID             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                      Threshold KPI Value                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Ingress KPIStamp                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Generic NSH KPI Encapsulation (Detection Mode)

Relevant fields in header that the FSN must implement:

- The O bit should not be set as we are operating on subscriber packets
- The C bit should be set indicating critical metadata exists
- The MD type must be set to 0x2
- The MD Class must be set to 0x0210 (General KPI Monitoring) as requested in Section 9. The stamp type is defined as per below:
  - Type = 0x00 Reserved.
  - Type = 0x01 Timestamp Detection
- The MSB of the Type field must be set to zero. Thus if a receiver along the path does not understand the KPIstamping protocol it will pass the packet transparently and not drop. This scheme allows for extensibility to the mechanism described in this document to other KPI collections and operations.

In the first header the SFC classifier may program a KPI threshold value. This is a value that when exceeded, requires the SF to set the C bit and insert the current SI value into the SI field. The KPI type is the type of KPI stamp inserted into the header as per section 9.

The flow ID is inserted into the header by the SFC classifier in order to correlate flow data in the KPIDB for offline analysis. The last two mandatory context headers are reserved for the KPIStamp. This is the KPI value at the chain ingress at the SFC classifier.

As an example operation, say we are using KPI type 0x01 (timestamp) when a service function (SFn) receives the packet it can compare current local timestamp (it first checks that it is synchronized to network PRC) with chain ingress timestamp to calculate the latency in the chain. If this value exceeds the timestamp threshold, it then sets the C bit inserts its SI and returns the NSH header to the KPIDB. This effectively tells the system that at SFn the packet violated the KPI threshold. All subsequent upstream SFs perform no NSH KPI operation as the flow has already been marked in violation via the C bit. Please refer to figure 9 for timestamp format.

When this occurs the SFC control plane system would then invoke the KPI extended mode, which uses a more sophisticated (and intrusive) method to isolate KPI violation root cause as described below.

Note: Whilst detection mode is a valuable tool for latency actions, we feel that it is not justified to build the logic into the KPI

system for QoS configuration. As QoS stamping is done infrequently and on a tiny percentage of user plane, it is more practical to use extended mode only for service chain QoS verification.

The generic NSH MD type 2 KPI Stamping header extended mode is shown below. This is the format we propose for performance monitoring of service chain issues with respect to QoS configuration and latency.

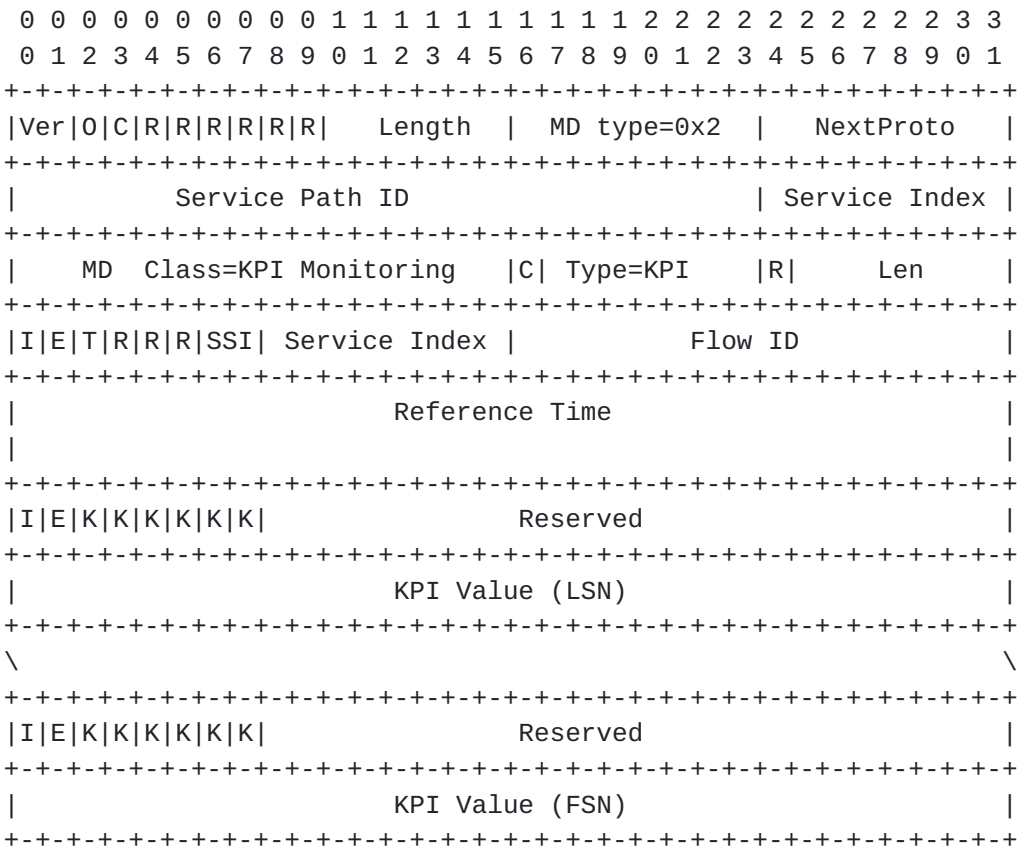

Figure 7: Generic KPI Encapsulation (Extended Mode)

As per section 9, we propose a new MD class 0x0210 to indicate KPI MD. Within this class we define 2 types for QoS and timestamp MD to be reported along the service chain. The K bits are KPI specific bits, for example, SYN for timestamping.

4.2. NSH Timestamping Encapsulation (Extended Mode)

The NSH timestamping encapsulation is shown below.

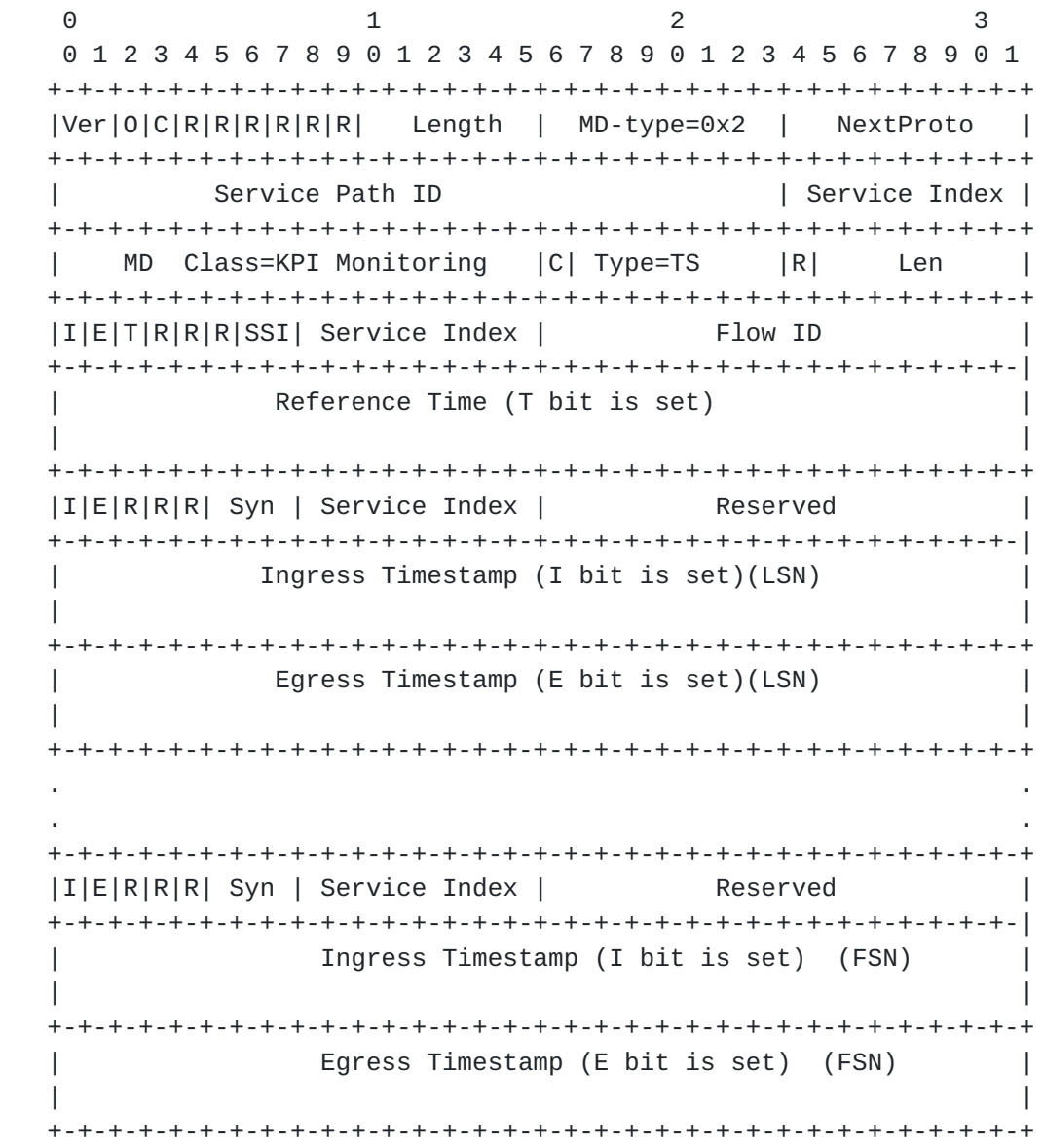

Figure 8: NSH Timestamp Encapsulation (Extended Mode)

Relevant fields in header that the FSN must implement:

- The O bit should not be set as we are operating on subscriber packets
- The C bit should be set indicating critical metadata exists
- The MD type must be set to 0x2
- The MD Class must be set to 0x0210 (General KPI Monitoring) as requested in Section 9. The stamp type is defined as per below:
  - Type = 0x00 Reserved
  - Type = 0x01 Timestamp Detection
  - Type = 0x02 Timestamp Extended
  - Type = 0x03 QoSStamp Extended
  - Type = 0x04 to 0x7F: Experimental
- The MSB of the Type field must be set to zero. Thus if a receiver along the path does not understand the KPIstamping protocol it will pass the packet transparently and not drop. This scheme allows for extensibility to the mechanism described in this document to other KPI collections and operations.

The FSN KPIstamp metadata starts with Stamping Configuration Header. This header contains the Stamp Service Index (SSI) field which must be set to one of the following values:

- 0x0 KPIstamp mode, no Service index specified in the Stamp Service Index field.
- 0x1 KPIUstamp Hybrid mode is selected, Stamp Service Index contains LSN Service index. This is used when PNFs or NSH-unaware SFs are used at the tail of the chain. If SSI=0x1, then the value in the type field informs the chain which SF should act as the LSN.

- 0x2 KPIstamp Specific mode is selected, Stamp Service Index contains the targeted Service Index. In this case the Stamp Service Index field indicates which SF is to be stamped. Both ingress and egress stamps are performed when the SI=SSI on the chain. For timestamping mode, the FSN will also apply the Reference Time and Ingress Timestamp. This will indicate the delay along the entire service chain to the targeted SF. This method may also be used as a light implementation to monitor end-to-end service chain performance whereby the targeted SF is the LSN. This is not applicable to QoSStamping mode.

The Flow ID is a unique 16 bit identifier written into the header by the classifier. This allow 65536 flows to be concurrently stamped on any given NSH service chain (SPI). Flow IDs are not written by subsequent SFs in the chain. The FSN may export monitored flow IDs to the KPIDB for correlation.

The E bit should be set if Egress stamp is requested.

The I bit should be set if Ingress stamp is requested.

The T bit should be set if Reference Time follows Stamping Configuration Header.

Reference Time is the wall clock of the FSN, and may be used for historical comparison of SC performance. If the FSN is not Level A synchronized (see Section 3.1) it should inform the SC over the SCP interface. The Reference Time is represented in 64-bit NTP format [RFC5905].

Each stamping Node adds stamping metadata which consist of Stamping Reporting Header and timestamps.

The E bit should be set if Egress stamp is reported.

The I bit should be set if Ingress stamp is reported.

With respect to timestamping mode, the Syn bits are an indication of the synchronization status of the node performing the timestamp and must be set to one of the following values:

- In Synch: 0x00

- In holdover: 0x01

- In free run: 0x02

o Out of Synch: 0x03

If the network node is out of synch or in free run no timestamp is applied by the node (but other timestamp MD is applied) and the packet is processed normally.

If FSN is out of synch or in free run timestamp request rejected and not propagated though the chain. The FSN should inform the SC in such an event over the SCP interface.

The outer service index value is copied into the stamp metadata to help cater for hybrid chains that's are a mix of VNFs and PNFs or through SFs that do not understand NSH. Thus if a flow transits through a PNF or an NSH-unaware node the delta in the inner service index between timestamps will indicate this.

The Ingress Timestamp and Egress Timestamp are represented in 64-bit NTP format [RFC5905]. The corresponding bits (I and E) reported in the Stamping Reporting Header of the node's metadata.

The 64-bit timestamp format [RFC5905] is presented below:

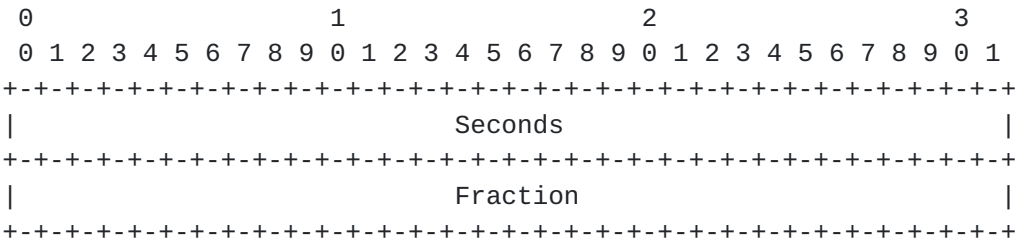

Figure 9: NTP [RFC5905] 64-bit Timestamp Format

4.3. NSH QoS Stamping Encapsulation (Extended Mode)

Packets have a variable QoS stack. That is for example the same payload IP can have a very different stack in the access part of the network to the core. This is most apparent in mobile networks where for example in an access circuit we would have 2 layers of infrastructure IP header (DSCP) - one transport-based and the other IPsec-based, in addition to multiple MPLS and VLAN tags. The same packet as it leaves the PGW Gi egress interface may be very much simplified in terms of overhead and related QoS fields.

Because of this variability we need to build extra meaning into the QoS headers - they are not for example all PTP timestamps of a fixed

length as in the case of timestamping, rather they are variable lengths and types. Also they can be changed on the underlay at any time without knowledge by the SFC system. Therefore each VNF must be able to ascertain and record its ingress and egress QoS configuration on the fly.

The suggested QoS type, lengths are as below. The type is 4 bits long.

Q Type(QT)	Value	Length	Comment
IVLAN	0x01	4 Bits	Ingress VLAN (PCP + DEI)
EVLAN	0x02	4 Bits	Egress VLAN
IQINQ	0x03	8 Bits	Ingress QinQ (2x PCP+DEI)
EQINQ	0x04	8 Bits	Egress QinQ
IMPLS	0x05	3 Bits	Ingress Label
EMPLS	0x06	3 Bits	Egress Label
IMPLS	0x07	6 Bits	2 Ingress Labels (2x EXP)
EMPLS	0x08	6 Bits	2 Egress Labels
IDSCP	0x09	8 Bits	Ingress DSCP
EDSCP	0x0A	8 Bits	Egress DSCP

For stacked headers such as MPLS and 802.1ad, we extract the QoS relevant data from the header and insert into one QoS value in order to be more efficient on packet size. This for MPLS we represent both EXP fields in one QoS value, and both 802.1p priority and drop precedence in one QoS value as indicated above.

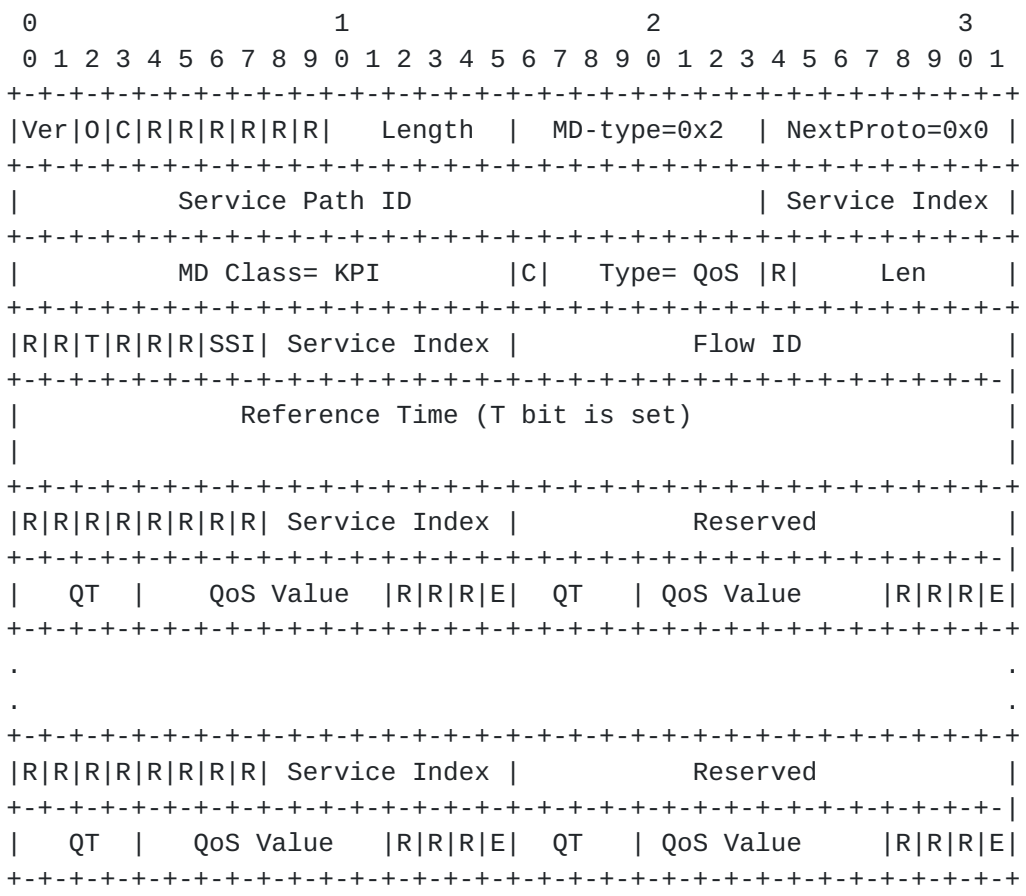

Figure 10: NSH QoS Configuration Encapsulation (Extended Mode)

The encapsulation above is very similar to that detailed in section 4.1 with the following exceptions

- I and E bits are not required as we wish to walk the full QoS stack at ingress and egress at every SF.

- Syn status bits are not required

- The QT (QoS Type) and QoS value are as outlined in the table above

- The E bit at the tail of each QoS context field indicates if this is the last egress QoS stamp for a given SF. This should coincide with SI=0 at the LSN, whereby the packet is truncated and the NSH MD sent to the KPIDB and the subscriber raw IP packet forwarded to the underlay next hop.

Note: It is possible to compress the frame structure to better utilize the header, but this would come at the expense of crossing byte boundaries. For ease of implementation, and that QoS stamping is applied on an extremely small subset of user plane traffic, we believe the above structure is a pragmatic compromise between header efficiency and ease of implementation.

5. Hybrid Models

A hybrid chain may be defined as a chain whereby there is a mix of NSH-aware and NSH-unaware SFs. This may be the case if some PNFs are used in the chain or if VNFs are used that do not support NSH.

Example 1. PNF in the middle

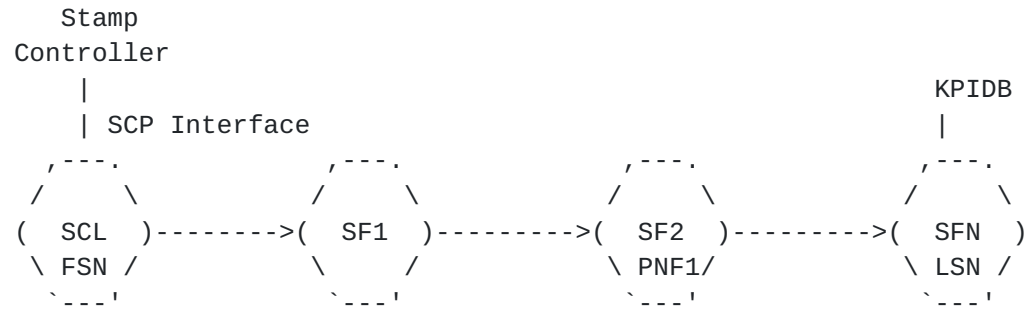

Figure 11: Hybrid chain with PNF in middle

In this example the FSN begins operation and sets the SI to 3, SF1 decrements this to 2 and passes the flow to an SFC proxy (not shown).

The proxy strips the NSH header and passes to the PNF. On receipt back from the PNF the Proxy decrements the SI and passes the packet onto the LSN with a SI=1.

After the LSN processes the traffic it knows it is the last node on the chain from the SI value and exports the entire NSH header and all metadata to the KPIDB. The payload is forwarded to the next hop on the underlay minus the NSH header. The TS information packet may be given a new SPI to act as a homing tag to transport the timestamp data back to the KPIDB.

Example 2. PNF at the end

Figure 12: Hybrid Chain with PNF at end

In this example the FSN begins operation and sets the SI to 3, the SSI field set to 0x1, and the type to 1. Thus when SF2 receives the packet with SI=1, it understands that it is expected to take on the role of the LSN as it is the last NSH-aware node in the chain.

5.1. Targeted VNF Stamp

For the majority of flows within the service chain, stamps (ingress, egress or both) will be carried out at each hop until the SI decrements to zero and the NSH header and Stamp MD is exported to the KPIDB. There may exist however the need to just test a particular VNF (perhaps after a scale out operation, software upgrade or underlay change for example). In this case the FSN should mark the NSH header as follows:

SSI field is set to 0x2. Type is set to the expected SI at the SF in question. When outer SI is equal to the SSI, stamps are applied at SF ingress and egress, and the NSH header and MD are exported to the KPIDB.

6. Fragmentation Considerations

The method described in this draft does not support fragmentation. The SC should return an error should a stamping request from an external system exceed MTU limits and require fragmentation.

Depending on the length of the payload and the type of KPIstamp and chain length, this will vary for each packet.

In most service provider architectures we would expect a SI << 10, and that may include some PNFs in the chain which do not add overhead. Thus for typical IMIX packet sizes we expect to able to perform timestamping on the vast majority of flows without fragmenting. Thus the classifier can have a simple rule to only allow KPIstamping on packet sizes less than 1200 bytes for example.

7. Security Considerations

The security considerations of NSH in general are discussed in [NSH].

The use of in-band timestamping, as defined in this document, can be used as a means for network reconnaissance. By passively eavesdropping to timestamped traffic, an attacker can gather information about network delays and performance bottlenecks.

The NSH timestamp is intended to be used by various applications to monitor the network performance and to detect anomalies. Thus, a man-in-the-middle attacker can maliciously modify timestamps in order to attack applications that use the timestamp values. For example, an attacker could manipulate the SFC classifier operation, such that it forwards traffic through 'better' behaving chains. Furthermore, if timestamping is performed on a fraction of the traffic, an attacker can selectively induce synthetic delay only to timestamped packets, causing systematic error in the measurements.

Similarly, if an attacker can modify QoS stamps, erroneous values may be imported into the KPIDB, resulting is further misconfiguration and subscriber QoE impairment.

An attacker that gains access to the SCP can enable time and QoS stamping for all subscriber flows, thereby causing performance bottlenecks, fragmentation, or outages.

As discussed in previous sections, NSH timestamping relies on an underlying time synchronization protocol. Thus, by attacking the time protocol an attack can potentially compromise the integrity of the NSH timestamp. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

8. Open Items for WG Discussion

- Specification and operation of SCP

o AOB

9. IANA Considerations

MD Class Allocation

MD classes are defined in [NSH].

IANA is requested allocate a new MD class value:

0x0210 KPI General Monitoring, stamping types and QoS types.

10. Contributors

This document originated as draft-browne-sfc-nsh-timestamp-00 and had the following co-authors and contributors. We would like to thank and recognize them and their contributions.

Yoram Moses

Technion

moses@ee.technion.ac.il

Brendan Ryan

Intel Corporation

brendan.ryan@intel.com

11. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

The authors would like to thank Ramki Krishnan and Anoop Ghanwani from Dell for their reviews and comments on this draft.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[NSH] Quinn, P., Elzur, U., Pignataro, C., "Network Service Header", draft-ietf-sfc-nsh-19 (work in progress), August 2017.

12.2. Informative References

[IEEE1588] IEEE TC 9 Instrumentation and Measurement Society, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5905] Mills, D., Martin, J., Burbank, J., Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, October 2014.

[Y.1731] ITU-T Recommendation G.8013/Y.1731, "OAM Functions and Mechanisms for Ethernet-based Networks", August 2015.

[Y.1564] ITU-T Recommendation Y.1564, "Ethernet service activation test methodology", March 2011.

[G.8261] ITU-T Recommendation G.8261/Y.1361, "Timing and synchronization aspects in packet networks", August 2013.

[G.8262] ITU-T Recommendation G.8262/Y.1362, "Timing characteristics of a synchronous Ethernet equipment slave clock", January 2015.

[G.8264] ITU-T Recommendation G.8264/Y.1364, "Distribution of timing information through packet networks", May 2014.

Authors' Addresses

Rory Browne
Intel
Dromore House
Shannon
Co.Clare
Ireland

Email: rory.browne@intel.com

Andrey Chilikin
Intel
Dromore House
Shannon
Co.Clare
Ireland

Email: andrey.chilikin@intel.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com