Independent Submission
Request for Comments: 8592
Category: Informational
ISSN: 2070-1721

R. Browne
A. Chilikin
Intel
T. Mizrahi
Huawei Network.IO Innovation Lab

May 2019

Key Performance Indicator (KPI) Stamping for the Network Service Header (NSH)

Abstract

This document describes methods of carrying Key Performance Indicators (KPIs) using the Network Service Header (NSH). These methods may be used, for example, to monitor latency and QoS marking to identify problems on some links or service functions.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8592.

Copyright Notice

Table of Contents

1. Introduction

The Network Service Header (NSH), as defined by [RFC8300], specifies a method for steering traffic among an ordered set of Service Functions (SFs) using an extensible service header. This allows for flexibility and programmability in the forwarding plane to invoke the appropriate SFs for specific flows.

The NSH promises a compelling vista of operational flexibility. However, many service providers are concerned about service and configuration visibility. This concern increases when considering 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 document describes generic methods to monitor and debug Service Function Chains (SFCs) in terms of latency and QoS marking of the flows within an SFC. These are referred to as "detection mode" and "extended mode" and are explained in Section 4.

The methods described in this document are compliant with hybrid architectures in which Virtual Network Functions (VNFs) and Physical Network Functions (PNFs) are freely mixed in the SFC. These methods also provide flexibility for monitoring the performance and configuration of an entire chain or parts thereof as desired. These methods are extensible to monitoring other Key Performance Indicators (KPIs). Please refer to [RFC7665] for an architectural context for this document.

The methods described in this document are not Operations, Administration, and Maintenance (OAM) protocols such as [Y.1731]. As such, they do not define new OAM packet types or operations. Rather, they monitor the SFC's performance and configuration for subscriber payloads and indicate subscriber QoE rather than out-of-band infrastructure metrics. This document differs from [In-Situ-OAM] in the sense that it is specifically tied to NSH operations and is not generic in nature.

2. Terminology

2.1. Requirements Language

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Definition of Terms

This section presents the main terms used in this document. This document also makes use of the terms defined in [RFC7665] and [RFC8300].

2.2.1. Terms Defined in This Document

First Stamping Node (FSN): The first node along an SFC that stamps packets using KPI stamping. The FSN matches each packet with a Stamping Controller (SC) flow based on (but not limited to) a stamping classification criterion such as transport 5-tuple coordinates.

Last Stamping Node (LSN): The last node along an SFC that stamps packets using KPI stamping. From a forwarding point of view, the LSN removes the NSH and forwards the raw IP packet to the next hop. From a control-plane point of view, the LSN reads all the metadata (MD) and exports it to a system performance statistics agent or repository. The LSN should use the NSH Service Index (SI) to indicate if an SF was at the end of the chain. The LSN may change the Service Path Identifier (SPI) to a preconfigured value so that the network underlay forwards the MD back directly to the KPI database (KPIDB) based on this value.

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

KPI stamping: The insertion of latency-related and/or QoS-related information into a packet using NSH MD.

Flow ID: A unique 16-bit identifier written into the header by the classifier. This allows 65536 flows to be concurrently stamped on any given NSH service chain.

QoS stamping: The insertion of QoS-related information into a packet using NSH MD.

Stamping Controller (SC): The central logic that decides what packets to stamp and how to stamp them. The SC instructs the classifier on how to build the parts of the NSH that are specific to KPI stamping.

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

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
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
RSP	Rendered Service Path
SC	Stamping Controller
SCL	Service Classifier
SCP	Stamping Control Plane
SF	Service Function
SFC	Service Function Chain
SI	Service Index
SSI	Stamp Service Index

TS	Timestamp
VLAN	Virtual Local Area Network
VNF	Virtual Network Function

3. NSH KPI Stamping: An Overview

A typical KPI-stamping architecture is presented in Figure 1.

```
       Stamping
      Controller
         |
         |                                                     KPIDB
         | SCP Interface                                        |
       ,---.             ,---.              ,---.              ,---.
      /     \           /     \            /     \            /     \
     (  SCL  )-------->(  SF1  )--------->(  SF2  )--------->(  SFn  )
      \ FSN /           \     /            \     /            \ LSN /
       `---'             `---'              `---'              `---'
```

Figure 1: Logical Roles in NSH KPI Stamping

The SC will be part of the SFC control-plane architecture, but it is described 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 the NSH-stamping policy into the service chain via the SCP interface.

The FSN will typically be part of the SCL but is called out as a separate logical entity for clarity.

The FSN is responsible for marking NSH MD fields; this tells nodes in the service chain how to behave in terms of stamping at the SF ingress, the SF egress, or both, or ignoring the stamp NSH MD completely.

The FSN also writes the Reference Time value, a (possibly inaccurate) estimate of the current time of day, into the header, allowing the "SPI:Flow ID" 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. The code and format of the error are specific to the protocol used between the FSN and SC; these considerations are out of scope.

SF1 and 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 SF itself and may be applied by a hardware device -- for example, as discussed in Section 3.3.

Note 2: Special cases exist where some of the SFs are NSH unaware. This is covered in Section 5.

The LSN should strip the entire NSH and forward the raw packet to the IP next hop as per [RFC8300]. The LSN also exports NSH-stamping information to the KPIDB for offline analysis; the LSN may export the stamping information of either (1) all packets or (2) a subset based on packet sampling.

In fully virtualized environments, the LSN is likely to be co-located with the SF that decrements the NSH SI to zero. Corner cases exist where this is not the case; see Section 5.

3.1. Prerequisites

Timestamping has its own set of prerequisites; however, these prerequisites are not required for QoS stamping. In order to guarantee MD accuracy, all servers hosting VNFs should be synchronized from a centralized stable clock. As it is assumed that PNFs do not timestamp (as this would involve a software change and a probable impact on throughput performance), 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 SF SHOULD have Level A synchronization and MAY have Level B synchronization.

Level A requires each platform (including the SC) to synchronize its system real-time clock to an NTP server. This is used to mark the MD in the chain, using the Reference Time field in the NSH KPI stamp header (Section 4.1). This timestamp is inserted into the NSH 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 time-of-day reference inserted into the KPIDB for performance monitoring and MD retrieval.

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

If an SF is not synchronized at the moment of timestamping, it should indicate its synchronization status in the NSH. 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. Indeed, the timestamp MD can indicate where a chain has been moved due to a resource starvation event as indicated in Figure 2, between VNF3 and VNF4 at time B.

```
     Delay
      |                                  v
      |                           v
      |                                  x
      |                           x             x = Reference Time A
      |                    xv                   v = Reference Time B
      |             xv
      |      xv
      |______|______|______|______|______|_____
         VNF1    VNF2   VNF3   VNF4   VNF5
```

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 a Reference Time for offline analysis, but this is not a hard requirement (they may be in holdover or free-run state, for example). Other SFs in the service chain do not need to be synchronized for QoS-stamping operations, as described below.

QoS stamping can be used to check the consistency of configuration across the entire chain or parts thereof. By adding all potential Layer 2 and Layer 3 QoS fields into a QoS sum at the SF ingress or egress, this allows quick identification of QoS mismatches across multiple Layer 2 / Layer 3 fields, which otherwise is a manual, expert-led consuming process.

```
   |
   |
   |                                  xy
   |                           xy          x = ingress QoS sum
   |                    xv                 v = egress QoS sum
   |             xv                        y = egress QoS sum mismatch
   |      xv
   |______|______|______|______|______|_____
         SF1    SF2    SF3    SF4    SF5
```

Figure 3: Flow QoS Consistency in a Service Chain

Referring to Figure 3, x, v, and y are notional sum values of the QoS marking 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, or DSCP(s), these values are used to compare the consistency of configuration from, for example, payload DSCP through overlay and underlay QoS settings in VLAN IEEE 802.1Q bits, MPLS bits, and infrastructure DSCPs.

Figure 3 indicates that, at SF4 in the chain, the egress QoS marking is inconsistent. That is, the ingress QoS settings do not match the egress. The method described here will indicate which QoS field(s) is inconsistent and whether this is ingress (where the underlay has incorrectly marked and queued the packet) or egress (where the SF has incorrectly marked and queued the packet.

Note that the SC must be aware of cases when an SF re-marks QoS fields deliberately and thus does not flag an issue for desired behavior.

3.2. Operation

KPI-stamping detection mode uses MD Type 2 as defined in [RFC8300]. This involves the SFC classifier stamping the flow at the chain ingress and no subsequent stamps being applied; rather, each upstream SF 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 4.2.

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:Flow ID". The SC should map these pairs to unique values presented as Flow IDs per service chain within the NSH TLV specified in this document (see Section 4). 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, by either the number of packets in the flow or a certain time duration.

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

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 a source IP address with a subscriber identity is outside the scope of this document and will vary by network application. For example, the method of association of a source IP address with an International Mobile Subscriber Identity (IMSI) will be different from how a Customer Premises Equipment (CPE) entity with a Network Address Translation (NAT) function may be chained in an enterprise NFV application.

3.2.2. SCP Interface

An SCP interface is required between the SC and the FSN or classifier. This interface is used to:

- Query the SFC classifier for a list of active chains and flows.

- Communicate which chains and flows to stamp. This can be a specific "SPI:Flow ID" combination or can include wildcards for monitoring subscribers across multiple chains or multiple flows within one chain.

- Instruct how the stamp should be applied (ingress, egress, both ingress and egress, or specific).

- Indicate when to stop stamping (after either a certain number of packets or a certain time duration).

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.

The exact specification of SCP is left for further study.

3.3. Performance Considerations

This document does not mandate a specific stamping implementation method; thus, NSH KPI stamping can be performed by either hardware mechanisms or 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, whilst 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 methods 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 a user plane every minute -- for example, to highlight any performance issues. Alternatively, the LSN may selectively export a subset of the KPI stamps 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 will have an impact on the performance of the chain itself whilst under investigation. This impact will be dependent on vendor implementations and is outside the scope of this document.

For QoS stamping, the methods described here are even less intrusive, as typically packets are only QoS stamped periodically (perhaps once per day) to check service chain configuration per QoS class.

4. NSH KPI-Stamping Encapsulation

KPI stamping uses NSH MD Type 0x2 for detection of anomalies and extended mode for root-cause analysis of KPI violations. These are further explained in this section.

The generic NSH MD Type 2 TLV for KPI stamping is shown below.

```
     0                   1                   2                   3
     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|U|    TTL    |   Length  |U|U|U|U|Type=2 | Next Protocol |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Service Path Identifier              | Service Index |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Metadata Class         |      Type     |U|    Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       Variable Length KPI Metadata header and TLV(s)          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Generic NSH KPI Encapsulation

Relevant fields in the header that the FSN must implement are as follows:

- The O bit must not be set.

- The MD type must be set to 0x2.

- The Metadata Class must be set to a value from the experimental range 0xfff6 to 0xfffe according to an agreement by all parties to the experiment.

- Unassigned bits: All fields marked "U" are unassigned and available for future use [RFC8300].

- The Type field may have one of the following values; the content of the Variable Length KPI Metadata header and TLV(s) field depends on the Type value:

  - Type = 0x01 (Det): Detection

  - Type = 0x02 (TS): Timestamp Extended

  - Type = 0x03 (QoS): QoS stamp Extended

The Type field determines the type of KPI-stamping format. The supported formats are presented in the following subsections.

4.1. KPI-Stamping Extended Encapsulation

The generic NSH MD Type 2 KPI-stamping header (extended mode) is shown in Figure 5. This is the format for performance monitoring of service chain issues with respect to QoS configuration and latency.

```
     0                   1                   2                   3
     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|U|    TTL    |   Length  |U|U|U|U|Type=2 | Next Protocol |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Service Path Identifier              | Service Index |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Metadata Class       |      Type     |U|    Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Variable Length KPI Configuration Header           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Variable Length KPI Value (LSN)              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \                                                               \
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Variable Length KPI Value (FSN)              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Generic KPI Encapsulation (Extended Mode)

As mentioned above, two types are defined under the experimental MD class to indicate the extended KPI MD: a timestamp type and a QoS-stamp type.

The KPI Encapsulation Configuration Header format is shown below.

```
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |K|K|T|K|K|K|K|K|   Stamping SI |           Flow ID             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Reference Time                      |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: KPI Encapsulation Configuration Header

The bits marked "K" are reserved for specific KPI type use and are described in the subsections below.

The T bit should be set if Reference Time follows the KPI Encapsulation Configuration Header.

The SSI (Stamping SI) contains the SI used for KPI stamping and is described in the subsections below.

The Flow ID is a unique 16-bit identifier written into the header by the classifier. This allows 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.

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], as presented in Figure 7:

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Seconds                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Fraction                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: NTP 64-Bit Timestamp Format (RFC 5905)

4.1.1. NSH Timestamping Encapsulation (Extended Mode)

The NSH timestamping extended encapsulation is shown below.

```
     0                   1                   2                   3
     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|U|U|U|U|U|U|   Length  |U|U|U|U|Type=2 |   NextProto   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Service Path ID                      | Service Index |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Metadata Class         |  Type=TS(2) |U|     Len     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |I|E|T|U|U|U|SSI|  Stamping SI  |           Flow ID             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |              Reference Time (T bit is set)                    |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |I|E|U|U|U| SYN |  Stamping SI  |         Unassigned            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |            Ingress Timestamp (I bit is set) (LSN)             |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Egress Timestamp (E bit is set) (LSN)             |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |I|E|U|U|U| SYN |  Stamping SI  |          Unassigned           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |                 Ingress Timestamp (I bit is set) (FSN)        |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Egress Timestamp (E bit is set) (FSN)         |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: NSH Timestamp Encapsulation (Extended Mode)

The FSN KPI stamp MD starts with the Stamping Configuration Header. This header contains the I, E, and T bits, and the SSI.

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

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

The SSI field must be set to one of the following values:

- 0x0: KPI stamp mode. No SI is specified in the Stamping SI field.
- 0x1: KPI stamp hybrid mode is selected. The Stamping SI field contains the LSN SI. 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 regarding which SF should act as the LSN.
- 0x2: KPI stamp Specific mode is selected. The Stamping SI field contains the targeted SI. In this case, the Stamping SI field indicates which SF is to be stamped. Both Ingress stamps and Egress stamps are performed when the SI=SSI in 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 QoS-stamping mode.

Each stamping node adds stamp MD that consists of the Stamping Reporting Header and timestamps.

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

The I bit should be set if the 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
- Out of synch: 0x03

If the platform hosting the SF is out of synch or in free run, no timestamp is applied by the node, and the packet is processed normally.

If the FSN is out of synch or in free run, the timestamp request is rejected and is not propagated through the chain. In such an event, the FSN should inform the SC over the SCP interface. Similarly, if the KPIDB receives timestamps that are out of order (i.e., a timestamp of an "N+1" SF is prior to the timestamp of an "N" SF), it should notify the SC of this condition over the SCP interface.

The outer SI value is copied into the stamp MD as the Stamping SI to help cater to hybrid chains that are a mix of VNFs and PNFs or through NSH-unaware SFs. Thus, if a flow transits through a PNF or an NSH-unaware node, the delta in the inner SI between timestamps will indicate this.

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

4.1.2. NSH QoS-Stamping Encapsulation (Extended Mode)

Packets have a variable QoS stack. For example, the same payload IP can have a very different stack in the access part of the network than the core. This is most apparent in mobile networks where, for example, in an access circuit we would have an infrastructure IP header (DSCP) composed of two layers -- one based on transport and the other based on IPsec -- in addition to multiple MPLS and VLAN tags. The same packet, as it leaves the Packet Data Network (PDN) Gateway 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 of variable lengths and types. Also, they can be changed on the underlay at any time without the knowledge of the SFC system. Therefore, each SF must be able to ascertain and record its ingress and egress QoS configuration on the fly.

The suggested QoS Type (QT) and lengths are listed below.

QoS Type	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	Two Ingress Labels (2x EXP)
EMPLS	0x08	6 Bits	Two 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 relevant QoS data from the header and insert it into one QoS value in order to be more efficient in terms of packet size. Thus, for MPLS, we represent both experimental bits (EXP) fields in one QoS value, and both 802.1p priority and drop precedence in one QoS value, as indicated above.

For stack types not listed here (for example, three or more MPLS tags), the SF would insert IMPLS/EMPLS several times, with each layer in the stack indicating EXP QoS for that layer.

```
     0                   1                   2                   3
     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|U|U|U|U|U|U|   Length  |U|U|U|U|Type=2 | NextProto=0x0 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Service Path ID                      | Service Index |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Metadata Class        |   Type=QoS(3) |U|     Len     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |U|U|T|U|U|U|SSI|  Stamping SI  |           Flow ID             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |              Reference Time (T bit is set)                    |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |U|U|U|U|U|U|U|U|  Stamping SI  |         Unassigned            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |   QT  |    QoS Value  |U|U|U|E|  QT   | QoS Value     |U|U|U|E|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                               .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |U|U|U|U|U|U|U|U|  Stamping SI  |          Unassigned           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
    |   QT  |   QoS Value   |U|U|U|E|  QT   | QoS Value     |U|U|U|E|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: NSH QoS Configuration Encapsulation (Extended Mode)

The encapsulation in Figure 9 is very similar to the encapsulation detailed in Section 4.1.1, with the following exceptions:

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

- SYN status bits are not required.

- The QT and QoS values are as outlined in the list 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, the NSH MD is sent to the KPIDB, and the subscriber’s raw IP packet is 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 so that QoS stamping is applied on an extremely small subset of user-plane traffic, we believe that the above structure is a pragmatic compromise between header efficiency and ease of implementation.

4.2. KPI-Stamping Encapsulation (Detection Mode)

The format of the NSH MD Type 2 KPI-stamping TLV (detection mode) is shown in Figure 10.

This TLV is used for KPI anomaly detection. Upon detecting a problem or an anomaly, it will be possible to enable the use of KPI-stamping extended encapsulations, which will provide more detailed analysis.

```
     0                   1                   2                   3
     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|U|    TTL    |   Length  |U|U|U|U|Type=2 | Next Protocol |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Service Path Identifier              | Service Index |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Metadata Class         | Type=Det(1)   |U|    Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   KPI Type    |      Stamping SI      |          Flow ID      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Threshold KPI Value                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       Ingress KPI stamp                       |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Generic NSH KPI Encapsulation (Detection Mode)

The following fields are defined in the KPIDB MD:

- KPI Type: This field determines the type of KPI stamp that is included in this MD. If a receiver along the path does not understand the KPI type, it will pass the packet on transparently and will not drop it. The supported values of KPI Type are:

  * 0x0: Timestamp

  * 0x1: QoS stamp

- Threshold KPI Value: In the first header, the SFC classifier may program a KPI threshold value. This is a value that, when exceeded, requires the SF to insert the current SI value into the SI field. The KPI type is the type of KPI stamp inserted into the header as per Figure 10.

- Stamping SI: This is the Service Identifier of the SF when the above threshold value is exceeded.

- Flow ID: The Flow ID is inserted into the header by the SFC classifier in order to correlate flow data in the KPIDB for offline analysis.

- Ingress KPI stamp: The last 8 octets are reserved for the KPI stamp. This is the KPI value at the chain ingress at the SFC classifier. Depending on the KPI type, the KPI stamp includes either a timestamp or a QoS stamp. If the KPI type is Timestamp, then the Ingress KPI stamp field contains a timestamp in 64-bit NTP timestamp format. If the KPI type is QoS stamp, then the format of the 64-bit Ingress KPI stamp is as follows.

```
     0                   1                   2                   3
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   QT  |     QoS Value |              Unassigned               |
    +-+-+-+-+-+-+-+-+-+-+-+-+                                       +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: QoS-Stamp Format (Detection Mode)

As an example operation, let’s say we are using KPI type 0x01 (Timestamp). When an SF (say SFn) receives the packet, it can compare the current local timestamp (it first checks that it is synchronized to the network’s PRC) with the chain Ingress Timestamp to calculate the latency in the chain. If this value exceeds the timestamp threshold, it then inserts its SI and returns the NSH to the KPIDB. This effectively tells the system that at SFn the packet violated the KPI threshold. Please refer to Figure 8 for the 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 the root cause of the KPI violation, as described below.

Note: Whilst detection mode is a valuable tool for latency actions, the authors feel that building the logic into the KPI system for QoS configuration is not justified. As QoS stamping is done infrequently and on a tiny percentage of the user plane, it is more practical to use extended mode only for service chain QoS verification.

5. Hybrid Models

A hybrid chain may be defined as a chain whereby there is a mix of NSH-aware and NSH-unaware SFs.

Figure 12 shows an example of a hybrid chain with a PNF in the middle.

```
      Stamping
     Controller
         |                                                      KPIDB
         | SCP Interface                                        |
       ,---.             ,---.              ,---.              ,---.
      /     \           /     \            /     \            /     \
     (  SCL  )-------->(  SF1  )--------->(  SF2  )--------->(  SFn  )
      \ FSN /           \     /            \ PNF1/            \ LSN /
       `---'             `---'              `---'              `---'
```

Figure 12: Hybrid Chain with PNF in Middle

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

The SFC proxy strips the NSH and passes the packet to the PNF. On receipt back from the PNF, the proxy decrements the SI and passes the packet to the LSN with SI=1.

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

Figure 13 shows an example of a hybrid chain with a PNF at the end.

```
     Stamping
    Controller
        |                                                      KPIDB
        | SCP Interface                                        |
      ,---.             ,---.              ,---.              ,---.
     /     \           /     \            /     \            /     \
    (  SCL  )-------->(  SF1  )--------->(  SF2  )--------->(  PNFN )
     \ FSN /           \     /            \ LSN /            \     /
      `---'             `---'              `---'              `---'
```

Figure 13: Hybrid Chain with PNF at End

In this example, the FSN begins its operation and sets the SI to 3. The SSI field is set to 0x1, and the type is set 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 Stamping

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

- The SSI field is set to 0x2.

- Type is set to the expected SI at the SF in question.

- When the outer SI is equal to the SSI, stamps are applied at the SF ingress and egress, and the NSH and MD are exported to the KPIDB.

6. Fragmentation Considerations

The methods described in this document do 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 KPI stamp and chain length, this will vary for each packet.

In most service provider architectures, we would expect SI << 10, which may include some PNFs in the chain that do not add overhead. Thus, for typical Internet Mix (IMIX) packet sizes [RFC6985], we expect to be able to perform timestamping on the vast majority of flows without fragmentation. Thus, the classifier can apply a simple rule that only allows KPI stamping on packet sizes less than 1200 bytes, for example.

7. Security Considerations

The security considerations for the NSH in general are discussed in [RFC8300].

In-band timestamping, as defined in this document, can be used as a means for network reconnaissance. By passively eavesdropping on 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 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-behaved" chains. Furthermore, if timestamping is performed on a fraction of the traffic, an attacker can selectively induce synthetic delay only to timestamped packets and can systematically trigger measurement errors.

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

An attacker that gains access to the SCP can enable timestamping 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 attacker 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. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

9.2. Informative References

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588, <https://standards.ieee.org/standard/1588-2008.html>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC6985] Morton, A., "IMIX Genome: Specification of Variable Packet Sizes for Additional Testing", RFC 6985, DOI 10.17487/RFC6985, July 2013, <https://www.rfc-editor.org/info/rfc6985>.

[Y.1731] ITU-T Recommendation G.8013/Y.1731, "Operations, administration and maintenance (OAM) functions and mechanisms for Ethernet-based networks", August 2015, <https://www.itu.int/rec/T-REC-G.8013/en>.

[G.8261] ITU-T Recommendation G.8261/Y.1361, "Timing and synchronization aspects in packet networks", August 2013, <https://www.itu.int/rec/T-REC-G.8261>.

[G.8262] ITU-T Recommendation G.8262/Y.1362, "Timing characteristics of a synchronous Ethernet equipment slave clock", November 2018, <https://www.itu.int/rec/T-REC-G.8262>.

[G.8264] ITU-T Recommendation G.8264/Y.1364, "Distribution of timing information through packet networks", August 2017, <https://www.itu.int/rec/T-REC-G.8264>.

[In-Situ-OAM] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., Bernier, D., and J. Lemon, "Data Fields for In-situ OAM", Work in Progress, draft-ietf-ippm-ioam-data-05, March 2019.

Acknowledgments

The authors gratefully acknowledge Mohamed Boucadair, Martin Vigoureux, and Adrian Farrel for their thorough reviews and helpful comments.

Contributors

This document originated as draft-browne-sfc-nsh-timestamp-00; the following people were coauthors of that draft. We would like to thank them and recognize them for their contributions.

Yoram Moses
Technion
Email: moses@ee.technion.ac.il

Brendan Ryan
Intel Corporation
Email: brendan.ryan@intel.com

Authors' Addresses

Rory Browne
Intel
Dromore House
Shannon
Co. Clare
Ireland

Email: rorybrowne@yahoo.com

Andrey Chilikin
Intel
Dromore House
Shannon
Co. Clare
Ireland

Email: andrey.chilikin@intel.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com