Network Working Group
Internet Draft
Intended status: Informational
Expires: February 2019

R. Browne
A. Chilikin
Intel
T. Mizrahi
Marvell

August 27, 2018

A Key Performance Indicators (KPI) Stamping for the Network Service Header (NSH) draft-browne-sfc-nsh-kpi-stamp-05

Abstract

This document describes an experimenal method of carrying Key Performance Indicators (KPIs) using the Network Service Header (NSH). This method 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 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 27, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction....2
2. Terminology....3
2.1. Requirement Language....3
2.2. Definition of Terms....3
2.2.1. Terms Defined in this Document....4
2.3. Abbreviations....4
3. NSH KPI Stamping: An Overview....6
3.1. Prerequisites....7
3.2. Operation....10
3.2.1. Flow Selection....10
3.2.2. SCP Interface....10
3.3. Performance Considerations....11
4. NSH KPI-stamping Encapsulation....12
4.1. KPI-stamping Extended Encapsulation....13
4.1.1. NSH Timestamping Encapsulation (Extended Mode)....15
4.1.2. NSH QoS-stamping Encapsulation (Extended Mode)....17
4.2. KPI-stamping Encapsulation (Detection Mode)....20
5. Hybrid Models....22
5.1. Targeted VNF Stamp....23
6. Fragmentation Considerations....23
7. Security Considerations....24
8. IANA Considerations....25
9. Contributors....25
10. Acknowledgments....25
11. References....26
11.1. Normative References....26
11.2. Informative References....26

1. Introduction

Network Service Header (NSH), as defined by [RFC8300], specifies a method for steering the traffic among an order 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.

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 a generic method to monitor and debug service function chains in terms of latency and QoS marking of the flows within a service function chain. Thus, it is possible to detect and debug performance issues and to detect and debug QoS misconfigurations on the chain.

The method described in the document is compliant with hybrid architectures in which Virtual Network Functions (VNFs) and Physical Network Functions (PNFs) are freely mixed in the service function chain. This method also provides flexibility to monitor the performance and configuration of an entire chain or part thereof as desired. This method is extensible to monitoring other KPIs. Please refer to [RFC7665] for an architectural context for this document.

The method described in this document is not an OAM protocol such as [Y.1731] or [Y.1564]. As such it does not define new OAM packet types or operation. Rather it monitors the service function chain performance and configuration for subscriber payloads and indicates subscriber QoE rather than out-of-band infrastructure metrics. This document differs from to [I-D.ippm.ioam] in the sense that it is specifically tied to NSH operation and not generic in nature.

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

This section presents the main terms used in this document. This document 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 a service function chain that stamps packets using KPI stamping. The FSN matches each packet with a Stamping Controller flow based on a stamping classification criterion such as transport 5-tuple coordiantes, but not limited to.

Last Stamping Node (LSN): The last node along a service function chain that stamps packets using KPI stamping. The LSN reads all the metadata and exports it to a system performance statistics agent or repository. The LSN should use the NSH Service Index (SI) to indicate if a SF was at the end of the chain. The LSN changes the Service Path Identifier (SPI) in order that the network underlay forwards the metadata back directly to the KPI database (KPIDB).

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

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

Flow ID: 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).

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

Stamping Controller (SC): The SC is the central logic that decides what packets to stamp and how. The SC instructs the classifier on how to build the NSH.

Stamp 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
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: An Overview

A typical KPI stamping architecture is presented in Figure 1.

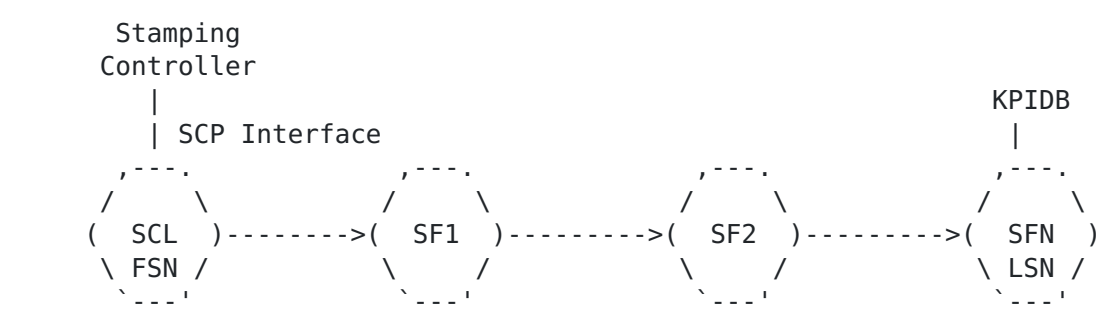

Figure 1: Logical roles in NSH KPI Stamping

The Stamping Controller (SC) will most probably be part of the SFC controller, but it is described separately in this document for clarity.

The SC is responsible for initiating start/stop stamp requests to the SCL or First Stamp Node (FSN), and also for distributing NSH stamping policy into the service chain via the Stamping Control Plane (SCP) interface.

The 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 which tells upstream nodes how to behave in terms of stamping at SF 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 {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 is 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, 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 Last Stamp Node (LSN) should strip the entire NSH header and forward the raw packet to the IP next hop as per [RFC8300]. 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 is likely to be co-located with the SF that decrements the NSH Service Index (SI) 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 (as this would involve a software change and probable throughput performance impact) 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 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 KPI-stamp header (Section 4.2). This timestamp is inserted to 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 metadata retrieval.

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 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 Rendered Service Path (RSP). Indeed the timestamp metadata can indicate where a chain has been moved due to a resource starvation event as indicated in Figure 2, between VNF 3 and VNF 4 at time B.

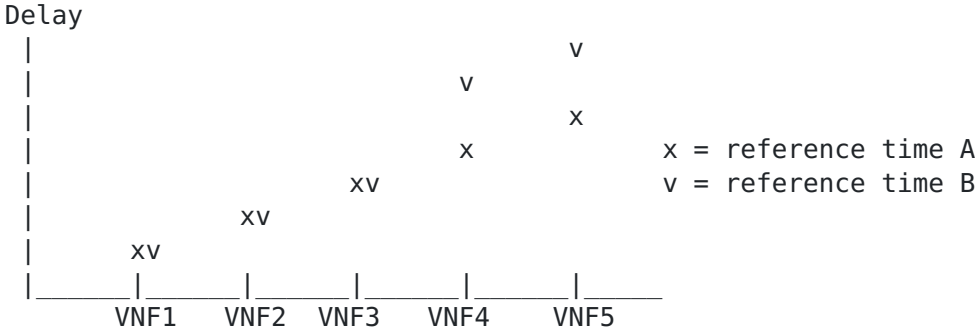

Figure 2: Flow performance in a service chain

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). Other SFs in the service chain 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. By adding all potential layer 2 and layer 3 QoS fields into a QoS sum at SF ingress or egress, this allows quick identification of QoS mismatches across multiple L2/L3 fields which otherwise is a manual, expert-led consuming process.

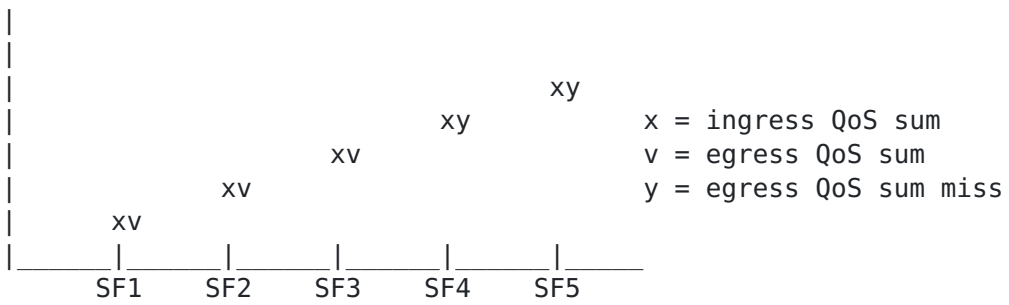

Figure 3: Flow QoS Consistency in a service chain

Referring to Figure 3, 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, 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, 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 (whereby 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 when a SF remarks QoS fields deliberately and thus does not flag an issue for desired behavior.

3.2. Operation

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

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:FlowID'. The SC should map these pairs to unique values presented as Flow IDs per service chain within the NSH TLV specified in this document. 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 a 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 source IP to 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 to IMSI will be different to how a CPE with NAT function may be chained in an enterprise NFV application.

3.2.2. SCP Interface

A Stamp Control Plane (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 include wildcards for monitoring subscribers across multiple chains or multiple flows within one chain.

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

- Indicate when to stop stamping, either after a certain number of packets or 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.

Exact specification of SCP is for further study.

3.3. Performance Considerations

This document 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 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 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 typically multiple packets in a flow are QoS stamped periodically (perhaps once per day) check one packet in a chain 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.

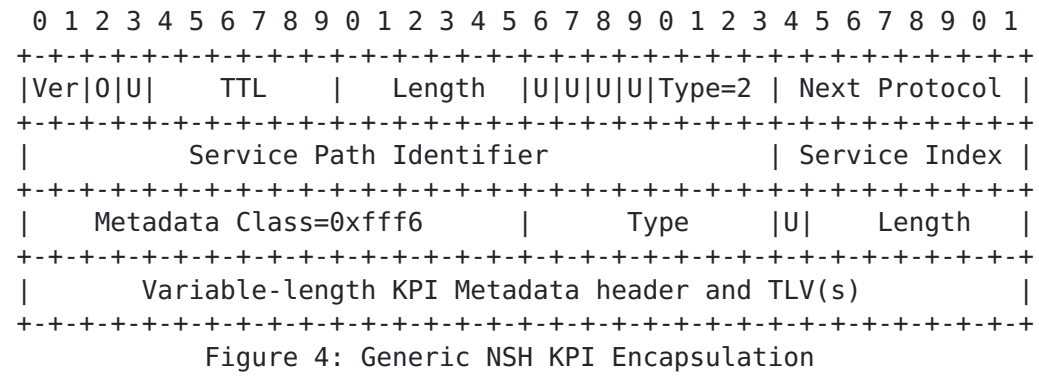

Figure 4: Generic NSH KPI Encapsulation

Relevant fields in header that the FSN must implement:

- The O bit must not be set.
- The MD type must be set to 0x2
- The MD Class must be set to 0xfff6.
- The Type field may have one of the following values; the content of "KPI metadata" depends on the type value:
  - Type = 0x01 Det: Detection
  - Type = 0x02 TS: Timestamp Extended

o 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 6. This is the format for performance monitoring of service chain issues with respect to QoS configuration and latency.

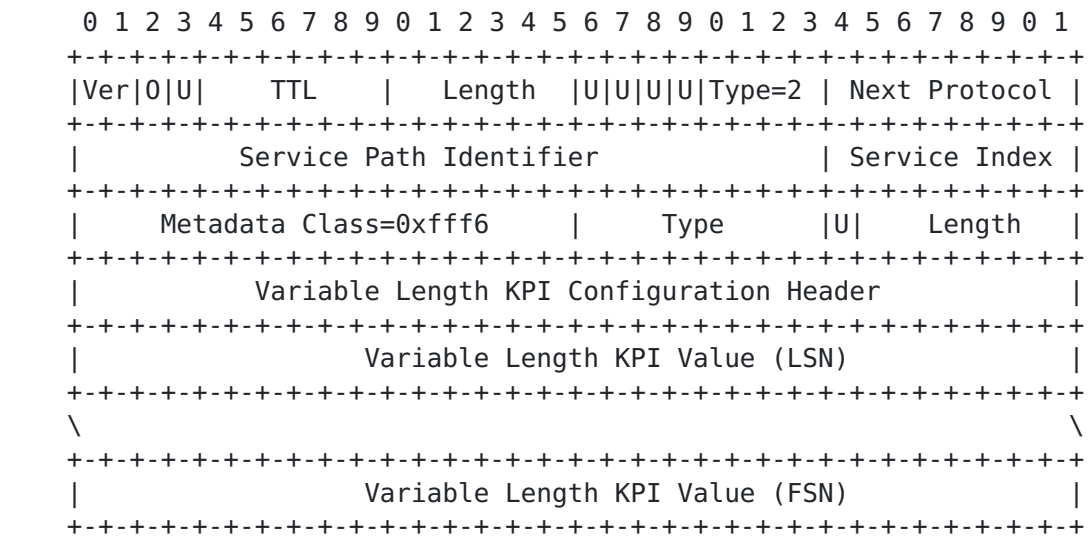

Figure 5: Generic KPI Encapsulation (Extended Mode)

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

The KPI Encapsulation Configuration Header format is shown below.

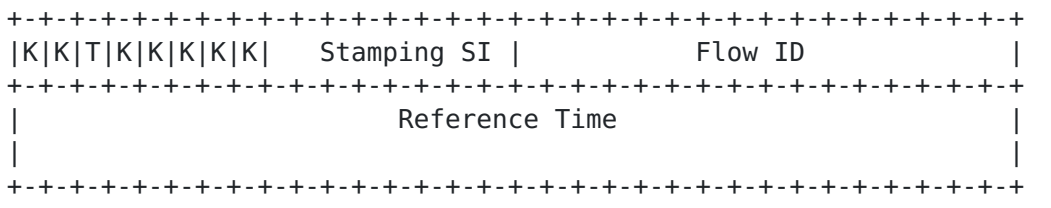

Figure 6: KPI Encapsulation Configuration Header

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

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

Stamping Service Index (Stamping SI) contains the Service Index used for KPI stamping and described in the corresponding 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] presented in Figure 8:

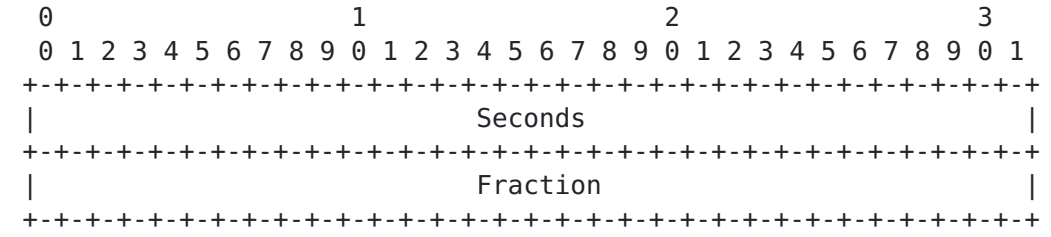

Figure 7: NTP [RFC5905] 64-bit Timestamp Format

4.1.1. NSH Timestamping Encapsulation (Extended Mode)

The NSH timestamping extended encapsulation is shown below.

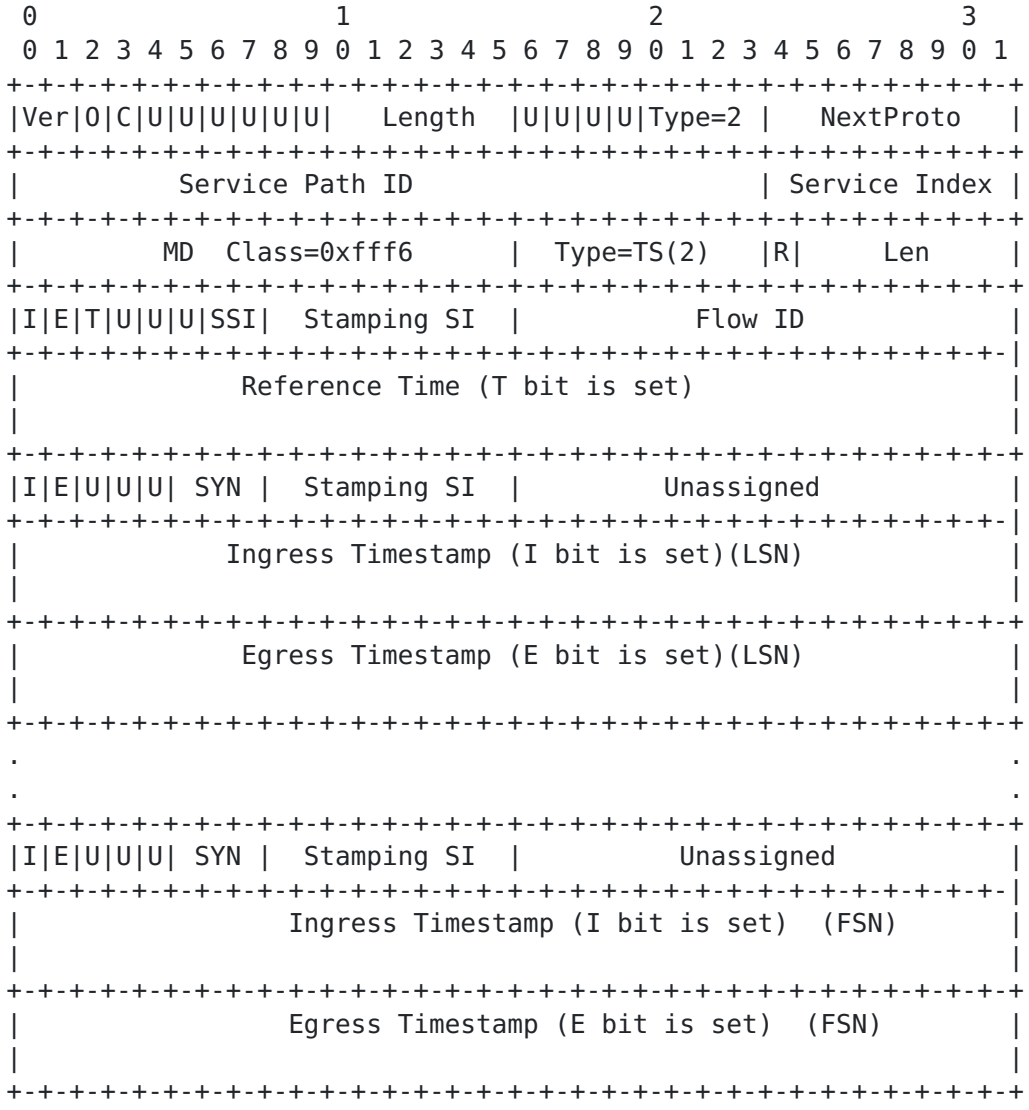

Figure 8: NSH Timestamp Encapsulation (Extended Mode)

The FSN KPI-stamp metadata starts with Stamping Configuration Header. This header contains the I, E, T bits and Stamp Service Index (SSI).

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

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

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

- 0x0 KPI-stamp mode, no Service index specified in the Stamp Service Index field.

- 0x1 KPI-stamp 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 KPI-stamp 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.

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

- 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 (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 as Stamping SI to help cater for 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 service index between timestamps will indicate this.

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

Timestamps are represented in 64-bit NTP [RFC5905] format, which is one of the recommended formats of [TS].

4.1.2. 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 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 variable lengths and types. Also they can be changed on the underlay at any time without knowledge by 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, lengths are as below. The type is 4 bits long.

QoS 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. Thus 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.

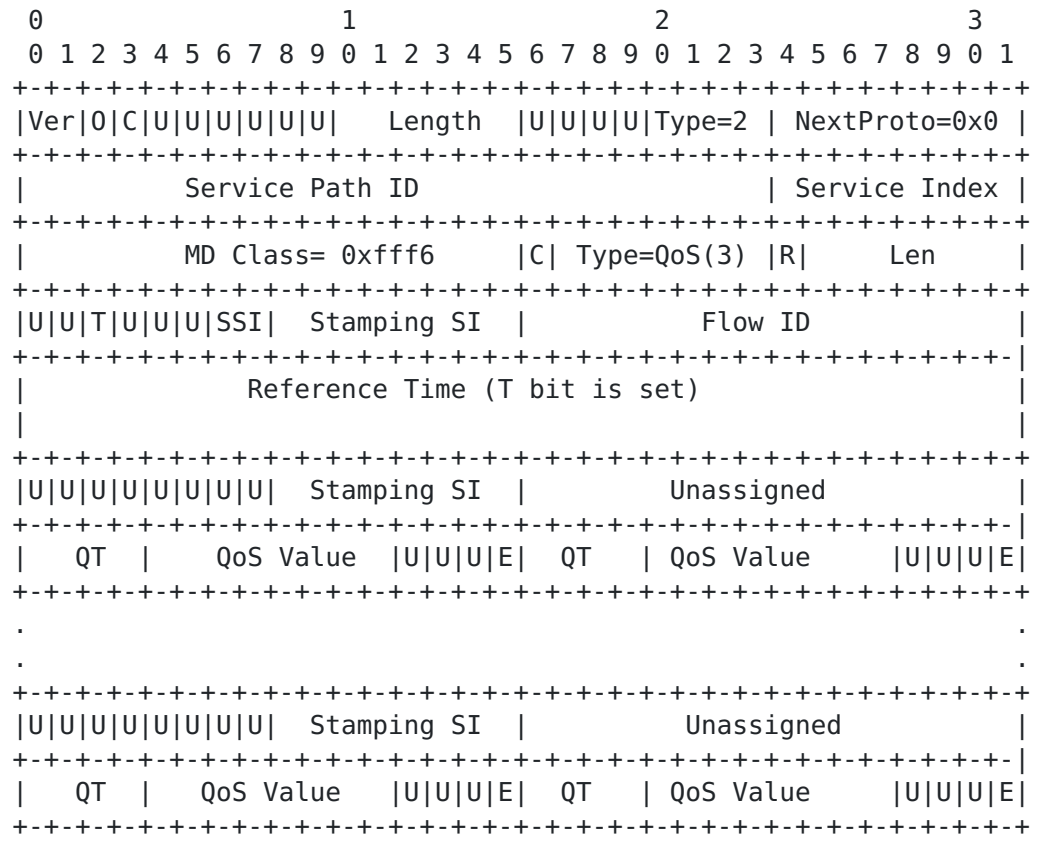

Figure 9: NSH QoS Configuration Encapsulation (Extended Mode)

The encapsulation in Figure 10 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 examine 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.

4.2. KPI-stamping Encapsulation (Detection Mode)

The format of the NSH MD type 2 KPI-stamping TLV (detection mode) is shown in Figure 11.

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.

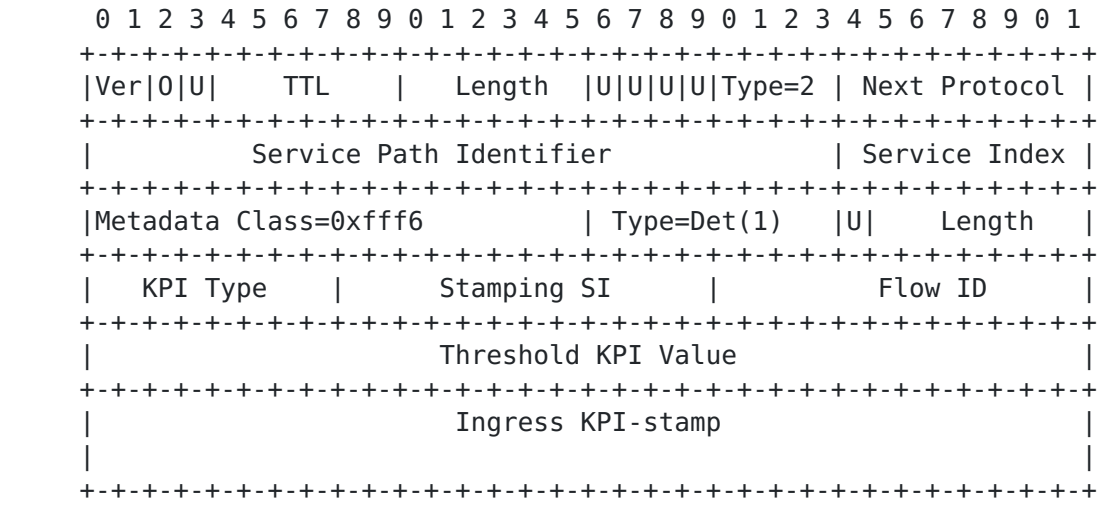

Figure 10: Generic NSH KPI Encapsulation (Detection Mode)

The following fields are defined in the KPI TSD metadata:

- KPI Type: determines the type of KPI-stamp that is included in this metadata field.
  If a receiver along the path does not understand the KPI Type it will pass the packet transparently and not drop.
  The supported values of the 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 section 9.

- Stamping SI: Service Identifier of the SF when the Threshold above 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 either includes 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.

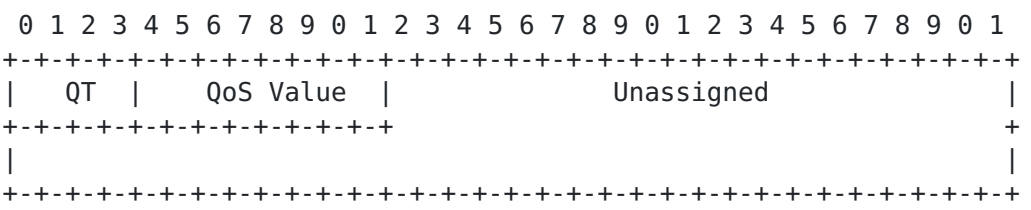

Figure 11: QoS-stamp Format (Detection Mode)

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 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 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, the authors 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.

5. Hybrid Models

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

Example 1. PNF in the middle

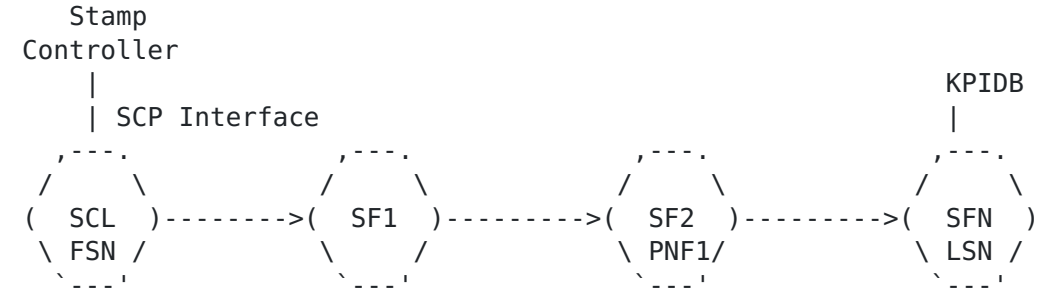

Figure 12: Hybrid chain with PNF in middle

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

The SFC proxy strips the NSH 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 and all metadata to the KPIDB. The payload is forwarded to the next hop on

the underlay minus the NSH. 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

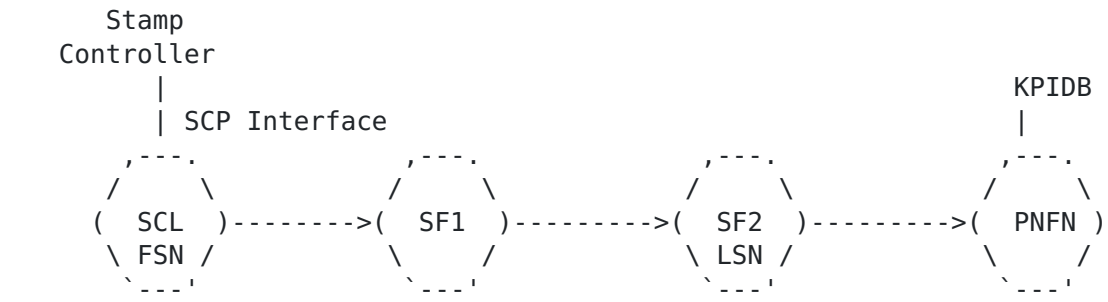

Figure 13: Hybrid Chain with PNF at end

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 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 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 and MD are exported to the KPIDB.

6. Fragmentation Considerations

The method described in this document 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 KPI-stamp 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 KPI-stamping on packet sizes less than 1200 bytes for example.

7. Security Considerations

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

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. IANA Considerations

IANA is requested allocate (register) new TLV types under the experimental MD class value 0xfff6:

- Type = 0x01: Detection
- Type = 0x02: Timestamp Extended
- Type = 0x03: QoS-stamp Extended

9. 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

10. 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 comments on this document. The authors also gratefully acknowledge Mohamed Boucadair for the thorough review and helpful comments.

11. References

11.1. Normative References

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

[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>.

[RFC8300] Quinn, P., Elzur, U., Pignataro, C., "Network Service Header (NSH)", RFC 8300, 2018.

11.2. 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.

[TS] Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", draft-ietf-ntp-packet-timestamps (work in progress), 2018.

[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.

[I-D.ippm.ioam]

Brockners, Bhandari et al. "Data Fields for In-situ OAM" draft-ietf-ippm-ioam-data-03 (work in progress), June 2018

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, 2066721 Israel

Email: tal.mizrahi.phd@gmail.com