sfc
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

F. Brockners
S. Bhandari
V. Govindan
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JMPC
T. Mizrahi
Marvell
D. Mozes
Mellanox Technologies Ltd.
P. Lapukhov
Facebook
R. Chang
Barefoot Networks
October 30, 2017

NSH Encapsulation for In-situ OAM Data
draft-brockners-sfc-ioam-nsh-00

Abstract

In-situ Operations, Administration, and Maintenance (OAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in the Network Service Header (NSH).

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Table of Contents

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than is being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the Network Service Header (NSH) [I-D.ietf-sfc-nsh]) encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. An implementation of IOAM which leverages NSH to carry the IOAM data is available from the FD.io open source software project [FD.io].

2. Conventions

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].

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

SFC: Service Function Chain

TLV: Type, Length, Value

3. IOAM data fields encapsulation in NSH

IOAM data fields are carried within the NSH header following NSH MDx metadata TLVs.

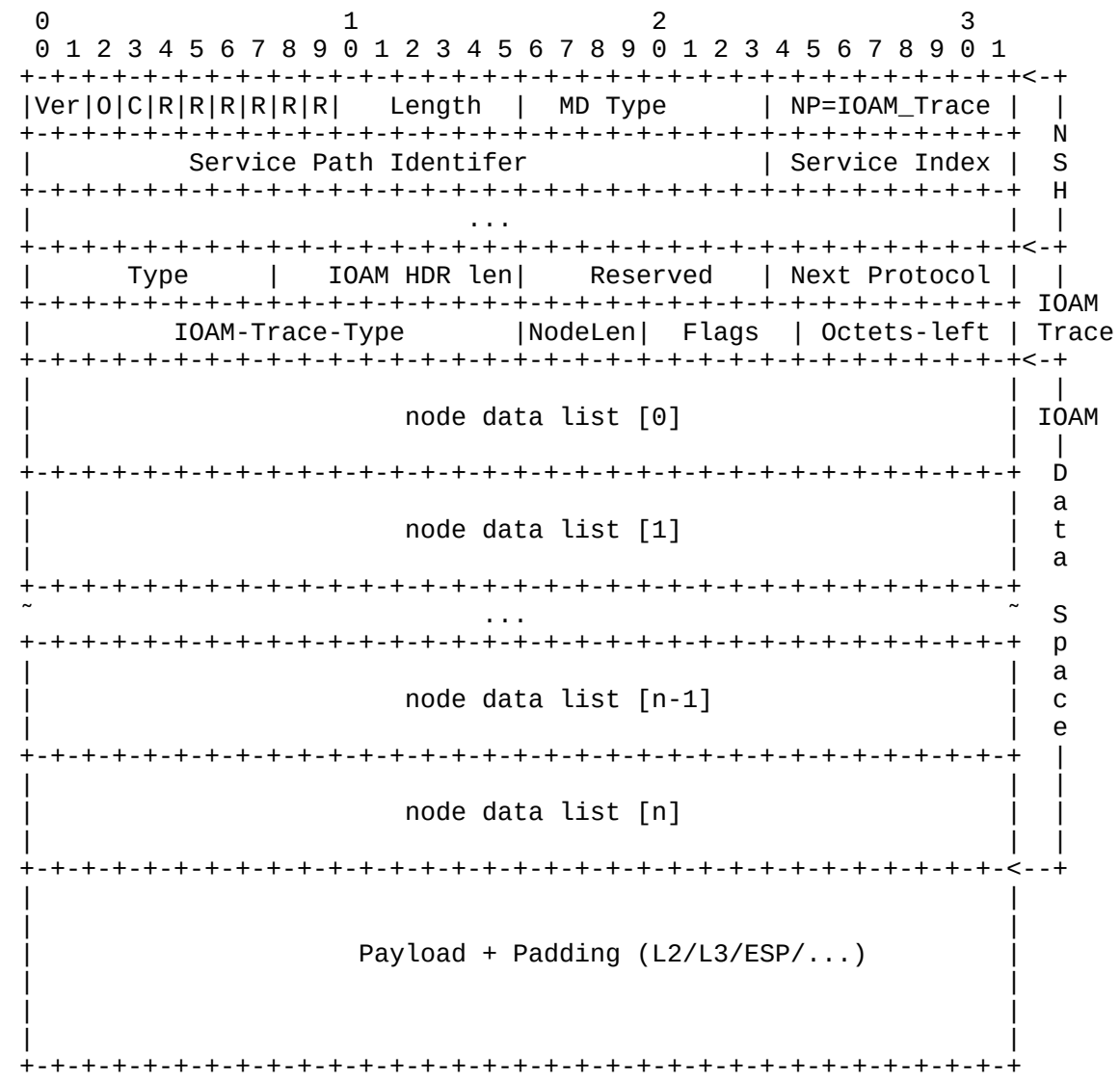
3.1. IOAM Trace Data in NSH

IOAM tracing data represents data that is inserted at nodes that a packet traverses. To allow for optimal implementations in both software as well as hardware forwarders, two different ways to encapsulate IOAM data are defined: "Pre-allocated" and "incremental". See [I-D.ietf-ippm-ioam-data] for details on IOAM tracing and the pre-allocated and incremental IOAM trace options.

The packet formats of the pre-allocated IOAM trace and incremental IOAM trace when transported in NSH are defined as below.

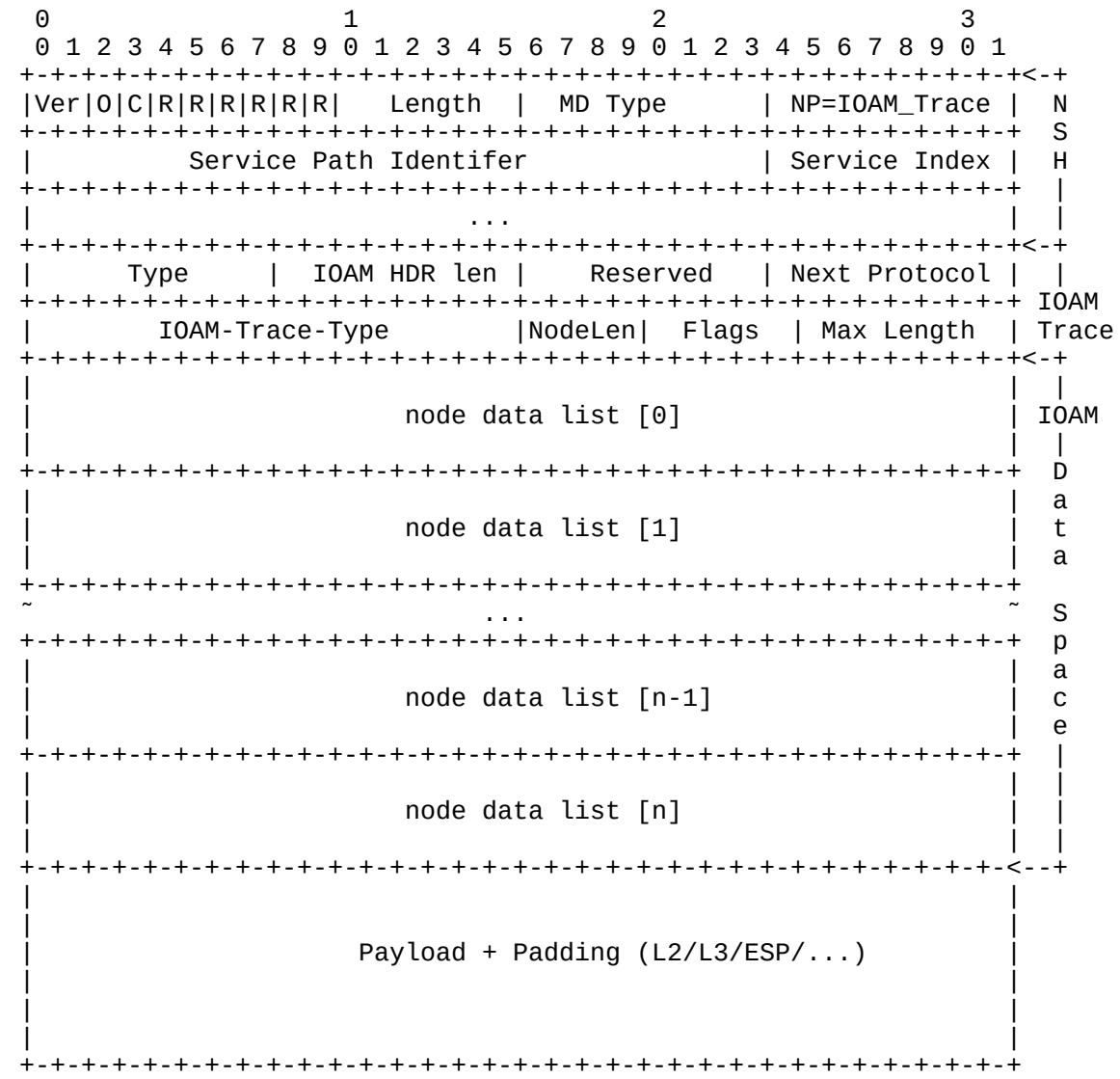
Note that in Service Function Chaining (SFC) [RFC7665], the Network Service Header (NSH) [I-D.ietf-sfc-nsh] already includes path tracing capabilities [I-D.penno-sfc-trace]. IOAM data fields for tracing complement the capabilities in NSH, in that IOAM data fields carry information complementary to information in NSH and benefit from the fact, that IOAM data fields use their own namespace. This allows intermediate nodes, which are not NSH hops to also process and update the IOAM data fields if configured to do so.

IOAM Trace header following NSH MDx header
(Pre-allocated IOAM trace):



IOAM Pre-allocated Trace Option Data MUST be 4-octet aligned:

IOAM Trace header following NSH MDx header
(Incremental IOAM trace):



IOAM Incremental Trace Option Data MUST be 4-octet aligned:

Next Protocol of NSH: TBD value for IOAM_Trace.

Type: 8-bit unsigned integer defining IOAM header type IOAM_TRACE_Preallocated or IOAM_Trace_Incremental are defined here.

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

IOAM-Trace-Type: 16-bit identifier of IOAM Trace Type as defined in [I-D.ietf-ippm-ioam-data] IOAM-Trace-Types.

Node Data Length: 4-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

Flags: 5-bit field as defined in [I-D.ietf-ippm-ioam-data].

Octets-left: 7-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

Maximum-length: 7-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

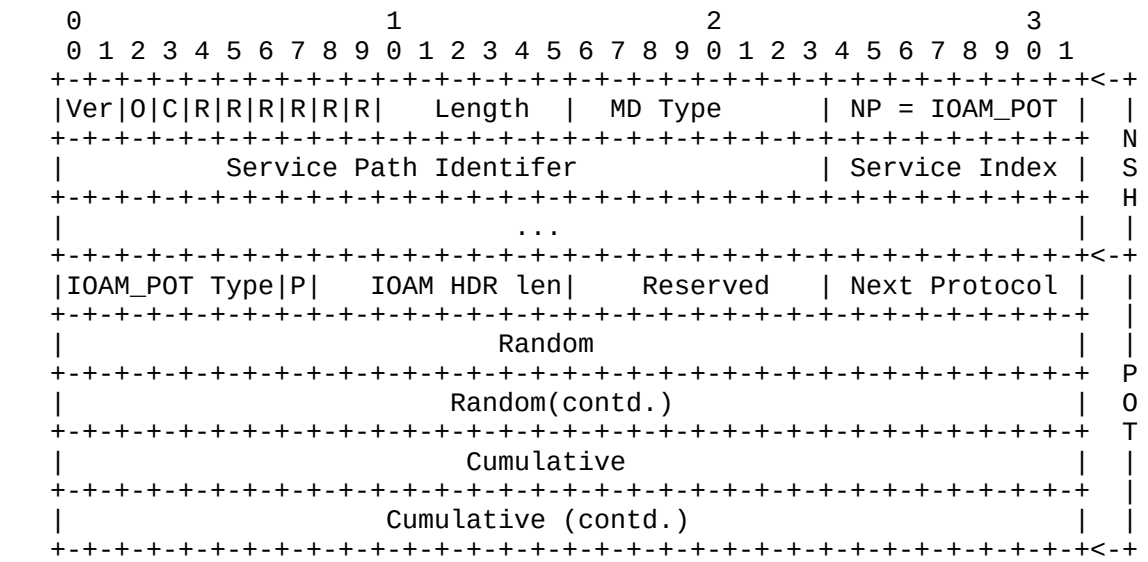
Node data List [n]: Variable-length field as defined in [I-D.ietf-ippm-ioam-data].

3.2. IOAM POT Data in NSH

IOAM proof of transit (POT, see [I-D.brockners-proof-of-transit]) offers a means to verify that a packet has traversed a defined set of nodes. In an administrative domain where IOAM is used, insertion of the IOAM data into the NSH header is enabled at the required nodes (i.e. at the IOAM encapsulating/decapsulating nodes) by means of configuration.

IOAM POT data fields are added as a TLV following NSH MDx metadata:

IOAM POT header following NSH MDx header:



Next Protocol of NSH: TBD value for IOAM_POT.

IOAM POT Type: 7-bit identifier of a particular POT variant that specifies the POT data that is to be included as defined in [I-D.ietf-ippm-ioam-data].

Profile to use (P): 1-bit as defined in [I-D.ietf-ippm-ioam-data] IOAM POT Option.

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

Random: 64-bit Per-packet random number.

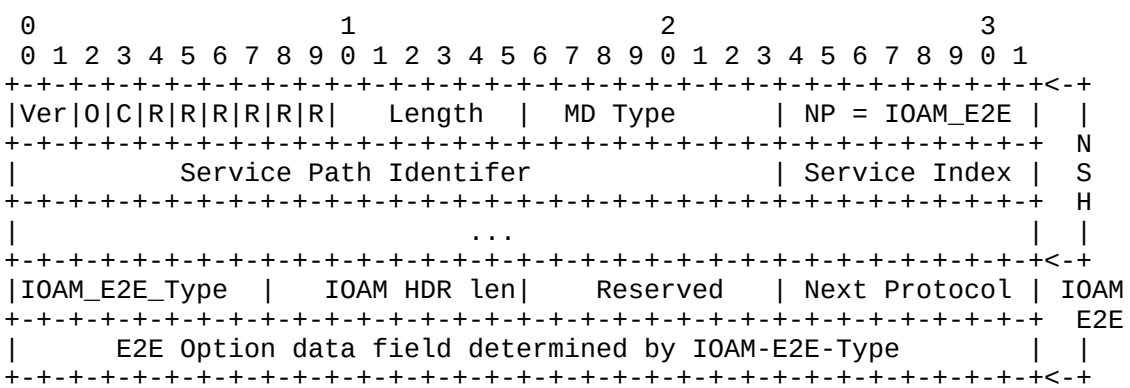
Cumulative: 64-bit Cumulative value that is updated by the Service Functions.

3.3. IOAM Edge-to-Edge Data in NSH

The IOAM edge-to-edge option is to carry data that is added by the IOAM encapsulating node and interpreted by the IOAM decapsulating node. The "Edge-to-Edge" capabilities (see [I-D.brockners-inband-oam-requirements]) of IOAM can be leveraged within NSH. In an administrative domain where IOAM is used, insertion of the IOAM data into the NSH header is enabled at the required nodes (i.e. at the IOAM encapsulating/decapsulating nodes) by means of configuration.

IOAM Edge-to-Edge data fields are added as a TLV following NSH MDx metadata:

IOAM E2E header following NSH MDx header:



Next Protocol of NSH: TBD value for IOAM_E2E.

IOAM E2E Type: 8-bit identifier of a particular E2E variant that specifies the IOAM E2E data that is to be included as defined in [I-D.ietf-ippm-ioam-data].

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

E2E Option data field: Variable length field as defined in [I-D.ietf-ippm-ioam-data] IOAM E2E Option.

4. Discussion of the encapsulation approach

This section is to support the working group discussion in selecting the most appropriate approach for encapsulating IOAM data fields in NSH.

An encapsulation of IOAM data fields in NSH should be friendly to an implementation in both hardware as well as software forwarders and support a wide range of deployment cases, including large networks that desire to leverage multiple IOAM data fields at the same time.

Hardware and software friendly implementation: Hardware forwarders benefit from an encapsulation that minimizes iterative look-ups of fields within the packet: Any operation which looks up the value of a field within the packet, based on which another lookup is performed, consumes additional gates and time in an implementation - both of which are desired to be kept to a minimum. This means that flat TLV structures are to be preferred over nested TLV structures. IOAM data fields are grouped into three option categories: Trace, proof-of-transit, and edge-to-edge. Each of these three options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure, but would rather carry the options as a serial sequence.

Total length of the IOAM data fields: The total length of IOAM data can grow quite large in case multiple different IOAM data fields are used and large path-lengths need to be considered. If for example an operator would consider using the IOAM trace option and capture node-id, app_data, egress/ingress interface-id, timestamp seconds, timestamps nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In case this particular deployment would have a maximum path length of 15 hops in the IOAM domain, then a maximum of 300 octets of IOAM data were to be encapsulated in the packet.

Two approaches for encapsulating IOAM data fields in NSH could be considered:

1. Encapsulation of IOAM data fields as "NSH MD Type 2" (see [I-D.ietf-sfc-nsh], section 2.5). Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by a type, with the different IOAM data fields being TLVs within this the particular option type. NSH MD Type 2 offers support for variable length meta-data. The length field is 6-bits, resulting in a maximum of 256 (2^6 x 4) octets.

2. Encapsulation of IOAM data fields using the "Next Protocol" field. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol".

The second option has been chosen here, because it avoids the additional layer of TLV nesting that the use of NSH MD Type 2 would result in. In addition, the second option does not constrain IOAM data to a maximum of 256 octets, thus allowing support for very large deployments.

5. IANA Considerations

IANA is requested to allocate protocol numbers for the following NSH "Next Protocols" related to IOAM:

Next Protocol	Description	Reference
x	IOAM_Trace	This document
y	IOAM_POT	This document
z	IOAM_E2E	This document

6. Security Considerations

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, and Andrew Yourtchenko for the comments and advice.

8. References

8.1. Normative References

[ETYPES] "IANA Ethernet Numbers", <https://www.iana.org/assignments/ethernet-numbers/ethernet-numbers.xhtml>.

[I-D.brockners-inband-oam-requirements]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., <>, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-00 (work in progress), September 2017.

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-04 (work in progress), April 2017.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[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>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <https://www.rfc-editor.org/info/rfc3232>.

8.2. Informative References

[FD.io] "Fast Data Project: FD.io", <https://fd.io/>.

[I-D.brockners-proof-of-transit]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Leddy, J., Youell, S., Mozes, D., and T. Mizrahi, "Proof of Transit", draft-brockners-proof-of-transit-03 (work in progress), March 2017.

[I-D.ietf-ippm-6man-pdm-option]
Elkins, N., Hamilton, R., and m. mackermann@bcbsm.com, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", draft-ietf-ippm-6man-pdm-option-13 (work in progress), June 2017.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-12 (work in progress), June 2017.

[I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.

[I-D.penno-sfc-trace]
Penno, R., Quinn, P., Pignataro, C., and D. Zhou, "Services Function Chaining Traceroute", draft-penno-sfc-trace-03 (work in progress), September 2015.

[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>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

David Mozes
Mellanox Technologies Ltd.

Email: davidm@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Remy Chang
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US