Network Working Group
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

Geneve encapsulation for In-situ OAM Data
draft-brockners-nvo3-ioam-geneve-00

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) 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 Geneve.

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 IOAM data fields are 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 Geneve [I-D.ietf-nvo3-geneve] encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data].

2. Conventions

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. Abbreviations

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

Geneve: Generic Network Virtualization Encapsulation

3. IOAM Data Field Encapsulation in Geneve

For encapsulating IOAM data fields into Geneve [I-D.ietf-nvo3-geneve] the different IOAM data fields are included in the Geneve header using tunnel options. IOAM data fields use a tunnel option class which includes the different types of IOAM data, including trace data, proof-of-transit data, and edge-to-edge data. In an administrative domain where IOAM is used, insertion of the IOAM tunnel option(s) in Geneve is enabled at the Geneve tunnel endpoints which also serve as IOAM encapsulating/decapsulating nodes by means of configuration. The Geneve header is defined in [I-D.ietf-nvo3-geneve]. IOAM specific fields for Geneve are defined in this document.

3.1. IOAM Trace Data in Geneve

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 encapsulated in Geneve are defined as below.

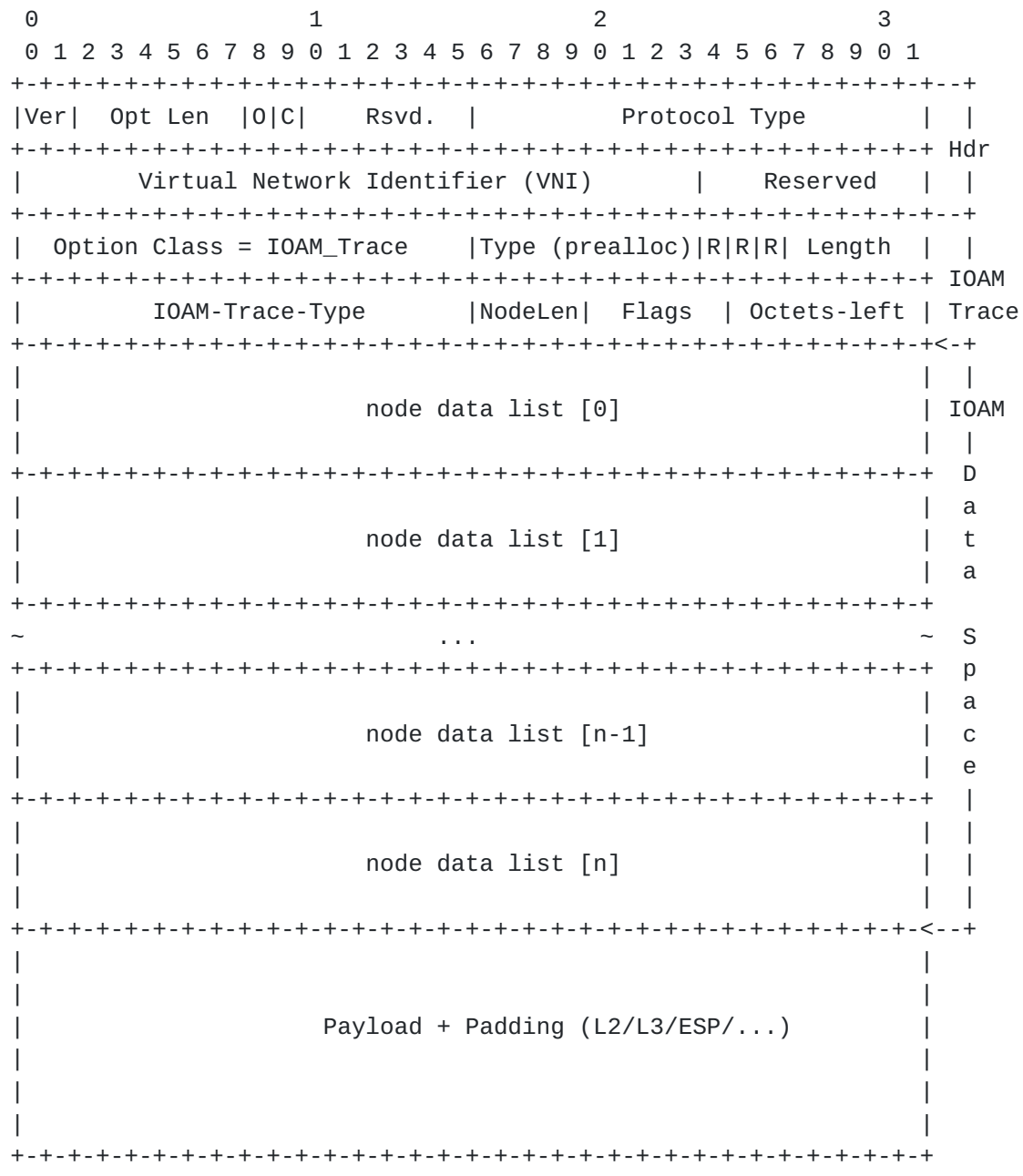

Figure 1: IOAM Pre-allocated Trace Option Format as a Geneve Tunnel Option

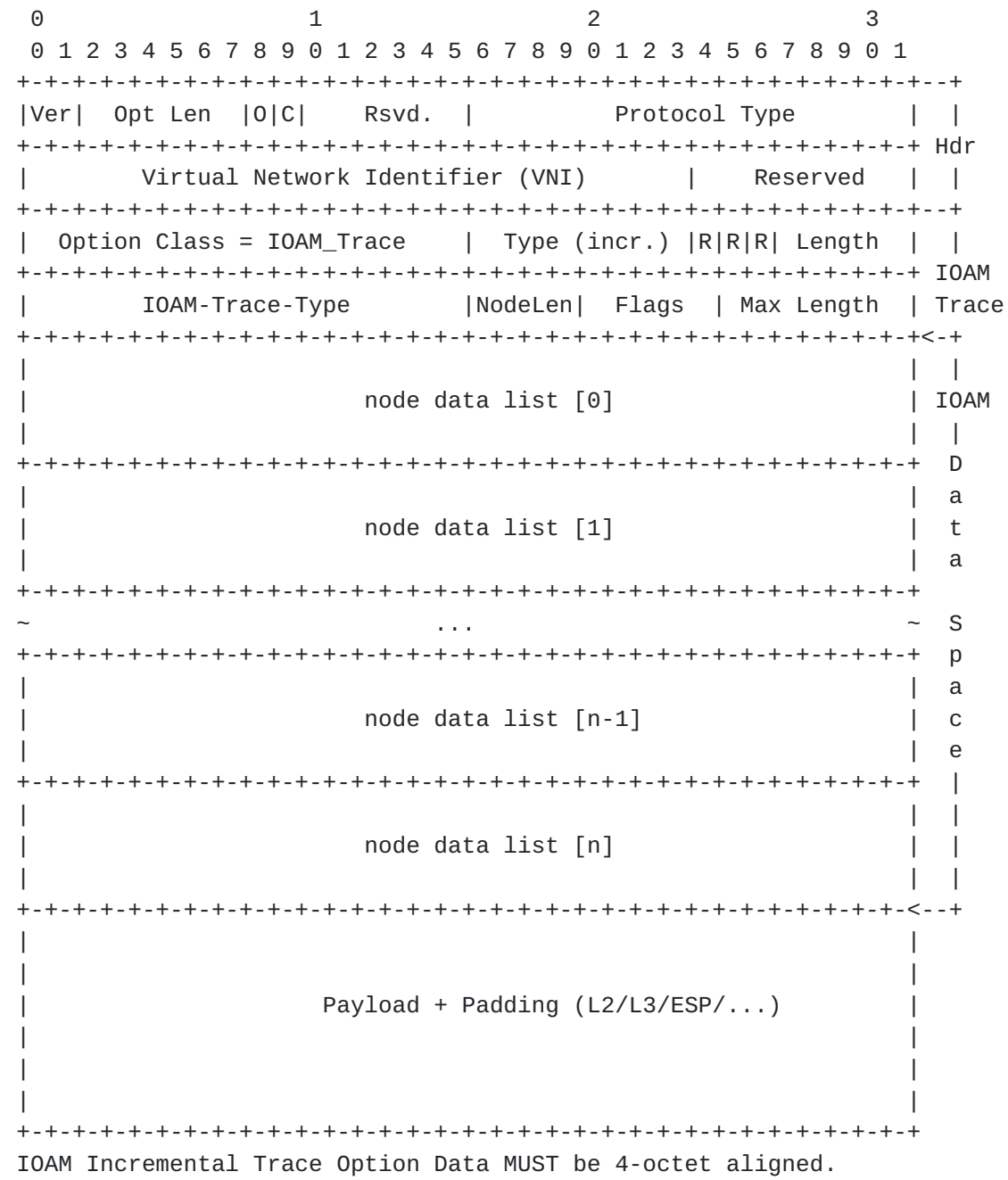

Figure 2: IOAM Incremental Trace Option Format as a Geneve Tunnel Option

The IOAM Trace header consists of 8 octets, as illustrated in Figure 1 and Figure 2. The first 4 octets are the Geneve Tunnel Option header [I-D.ietf-nvo3-geneve]. The next 4 octets are the trace option header; its format is defined in [I-D.ietf-ippm-ioam-data], and is described here for the sake of clarity.

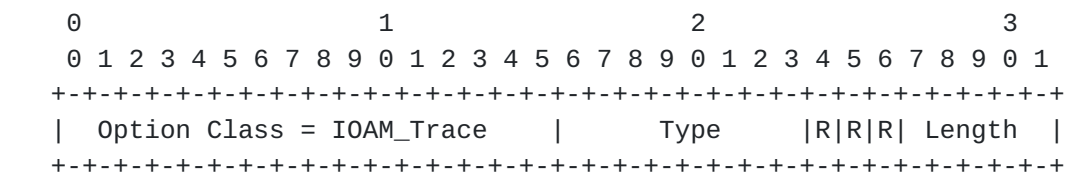

Figure 3: Geneve Tunnel Option for IOAM

The fields of the Geneve tunnel option are as follows:

Option Class: 16-bit unsigned integer that determines the IOAM option class. The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

The fields of the trace option header [I-D.ietf-ippm-ioam-data] are as follows:

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 Geneve

IOAM proof of transit (POT, see also [I-D.brockners-proof-of-transit]) offers a means to verify that a packet has traversed a defined set of nodes. IOAM POT data fields are encapsulated in Geneve as follows:

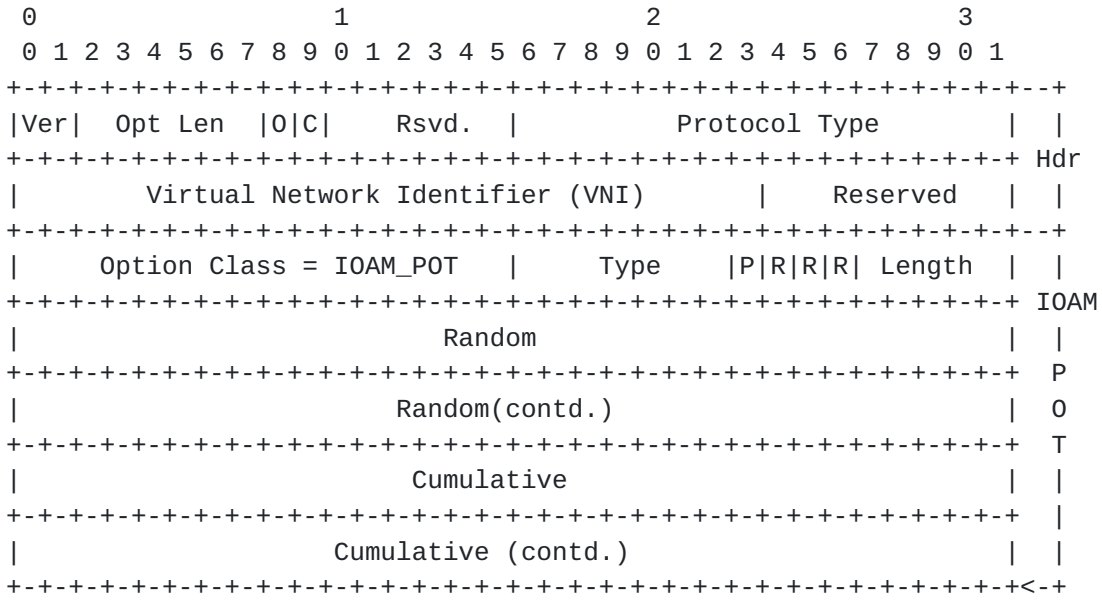

Figure 4: IOAM POT Header Following using a Geneve Tunnel Option

The first 4 octets of the IOAM POT are the Geneve tunnel option header (Figure 5), which includes the following fields:

Option Class: 16-bit unsigned integer that determines the IOAM_POT option class.The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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.

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

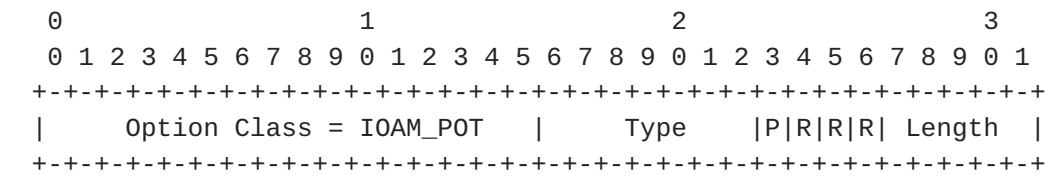

Figure 5: Geneve Tunnel Option for IOAM POT

The rest of the fields in the POT option [I-D.ietf-ippm-ioam-data] are as follows:

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 Geneve

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. IOAM specific fields to encapsulate IOAM Edge-to-Edge data fields are defined as follows:

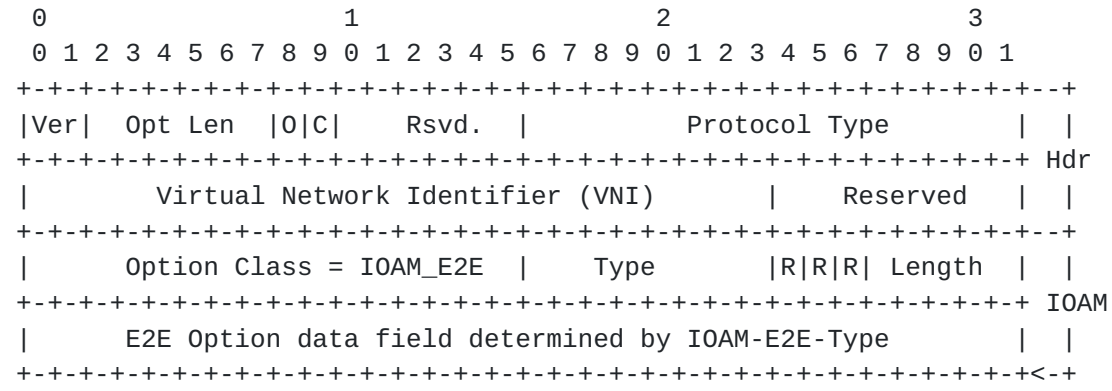

Figure 6: IOAM Edge-to-Edge using a Geneve Tunnel Option

The first 4 octets of the IOAM E2E option are the Geneve tunnel option header (Figure 5), which includes the following fields:

Option Class 16-bit unsigned integer that determines the IOAM_E2E option class.The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

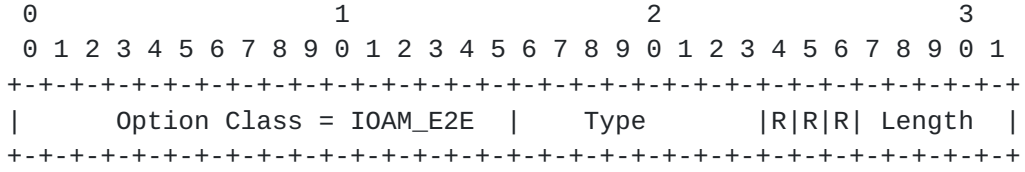

Figure 7: Geneve Tunnel Option for IOAM E2E

The rest of the E2E option [I-D.ietf-ippm-ioam-data] consists of:

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 Geneve.

An encapsulation of IOAM data fields in Geneve 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.

Concerns with the current encapsulation approach:

Hardware support: Using Geneve tunnel options to encapsulate IOAM data fields leads to a nested TLV structure. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) represents a type, with the different IOAM data fields being TLVs within this the particular option type. Nested TLVs require iterative look-ups, a fact that creates potential challenges for implementations in hardware. It would be desirable to offer a way to encapsulate IOAM in a way that keeps TLV nesting to a minimum.

Length: Geneve tunnel option length is a 5-bit field in the current specification [I-D.ietf-nvo3-geneve] resulting in a maximum option length of 128 (2^5 x 4) octets which constrains the use of IOAM to either small domains or a few IOAM data fields only. Support for large domains with a variety of IOAM data fields would be desirable.

5. IANA Considerations

IANA is requested to allocate a Geneve "option class" numbers for the following IOAM types:

Option Class	Description	Reference
x	IOAM_Trace	This document
y	IOAM_POT	This document
z	IOAM_E2E	This document

6. Security Considerations

The security considerations of Geneve are discussed in [I-D.ietf-nvo3-geneve], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

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-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-05 (work in progress), September 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.

[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