

ippm  
Internet-Draft  
Intended status: Informational  
Expires: 2 September 2024

F. Brockners, Ed.  
Cisco  
S. Bhandari  
Thoughtspot  
V. Govindan  
Cisco  
C. Pignataro, Ed.  
NC State University  
H. Gredler  
RtBrick Inc.  
J. Leddy

S. Youell  
JPMC

T. Mizrahi, Ed.  
Huawei  
A. Kfir  
B. Gafni  
Nvidia

P. Lapukhov  
Facebook

M. Spiegel

Barefoot Networks, an Intel company  
1 March 2024

VXLAN-GPE Encapsulation for In Situ OAM (IOAM) Data  
draft-brockners-ippm-ioam-vxlan-gpe-05

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 VXLAN-GPE.

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 <https://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 2 September 2024.

Copyright Notice

Copyright (c) 2024 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- 1. Introduction . . . . . 3
- 2. Conventions . . . . . 3
  - 2.1. Requirements Language . . . . . 3
  - 2.2. Abbreviations . . . . . 3
- 3. IOAM Data Field Encapsulation in VXLAN-GPE . . . . . 3
- 4. Considerations . . . . . 5
  - 4.1. Discussion of the Encapsulation Approach . . . . . 5
  - 4.2. IOAM and the Use of the VXLAN-GPE O-bit . . . . . 6
  - 4.3. Transit Devices . . . . . 6
- 5. IANA Considerations . . . . . 6
  - 5.1. VXLAN-GPE Next Protocol Value . . . . . 6
  - 5.2. LISP-GPE Next Protocol Value . . . . . 7
- 6. Security Considerations . . . . . 7
- 7. Acknowledgements . . . . . 7
- 8. References . . . . . 7
  - 8.1. Normative References . . . . . 7
  - 8.2. Informative References . . . . . 8
- Authors' Addresses . . . . . 8

## 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 being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe] encapsulation. The IOAM data fields are defined in [RFC9197]. An implementation of IOAM which leverages VXLAN-GPE to carry the IOAM data is available from the FD.io open-source software project [FD.io].

## 2. Conventions

### 2.1. Requirements 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

OAM:

Operations, Administration, and Maintenance

VXLAN-GPE:

Virtual eXtensible Local Area Network, Generic Protocol Extension

## 3. IOAM Data Field Encapsulation in VXLAN-GPE

VXLAN-GPE is defined in [I-D.ietf-nvo3-vxlan-gpe]. IOAM data fields are carried in VXLAN-GPE using a next protocol value of TBD\_IOAM. An IOAM header is added containing the different IOAM data fields defined in [RFC9197]. In an administrative domain where IOAM is used, insertion of the IOAM header in VXLAN-GPE is enabled at the VXLAN-GPE tunnel endpoints, which also serve as IOAM encapsulating/decapsulating nodes by means of configuration.

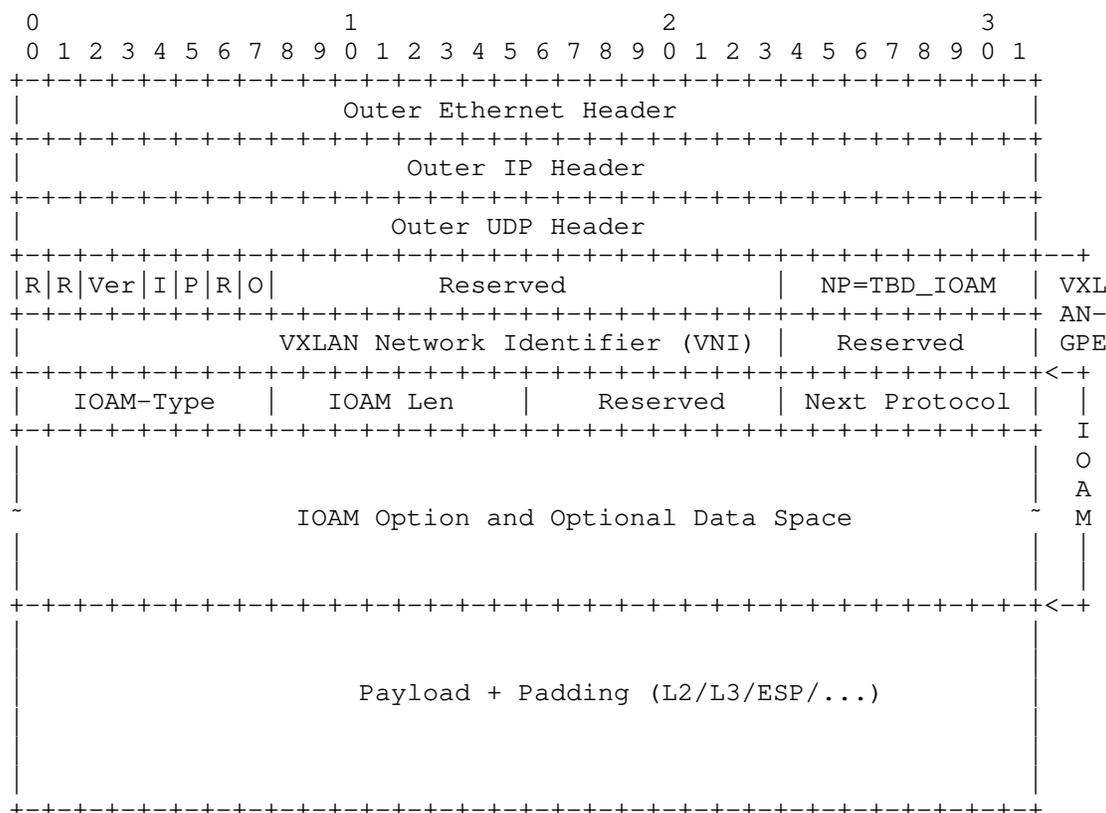


Figure 1: IOAM Data Encapsulation in VXLAN-GPE

The VXLAN-GPE header and fields are defined in [I-D.ietf-nvo3-vxlan-gpe]. The VXLAN-GPE Next Protocol value for IOAM is TBD\_IOAM.

The IOAM related fields in VXLAN-GPE are defined as follows:

**IOAM-Type:**

8-bit field defining the IOAM Option type, as defined in Section 7.2 of [RFC9197].

**IOAM Len:**

8-bit unsigned integer. Length of the IOAM protocol message shim, expressed in 4-octet units not including the first 4 octets.

**Reserved:**

8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.

**Next Protocol:**

8-bit unsigned integer that determines the type of header following IOAM protocol. The value is from the IANA registry setup for VXLAN-GPE Next Protocol defined in [I-D.ietf-nvo3-vxlan-gpe].

**IOAM Option and Optional Data Space:**

IOAM option header and data is present as specified by the IOAM-Type field and is defined in Section 4 of [RFC9197].

Multiple IOAM options MAY be included within the VXLAN-GPE encapsulation. For example, if a VXLAN-GPE encapsulation contains two IOAM options before a data payload, the Next Protocol field of the first IOAM option will contain the value of TBD\_IOAM, while the Next Protocol field of the second IOAM option will contain the VXLAN-GPE "Next Protocol" number indicating the type of the data payload.

#### 4. Considerations

This section summarizes a set of considerations on the overall approach taken for IOAM data encapsulation in VXLAN-GPE, as well as deployment considerations.

##### 4.1. 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 VXLAN-GPE.

An encapsulation of IOAM data fields in VXLAN-GPE should be friendly to an implementation in both hardware as well as software forwarders. Hardware forwarders benefit from an encapsulation that minimizes iterative lookups 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 and would rather carry the options as a serial sequence.

Two approaches for encapsulating IOAM data fields in VXLAN-GPE could be considered:

1. Use a single GPE protocol type for all IOAM types: IOAM would receive a single GPE protocol type code point. A "sub-type" field would then specify what IOAM options type (e.g., trace, proof-of-transit, and edge-to-edge) is carried.
2. Use one GPE protocol type per IOAM options type: Each IOAM data field option (e.g., trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol", i.e. each IOAM options type becomes its own GPE protocol type with a dedicated code point. This implies that in case additional IOAM option types would be added in the future, additional GPE protocol type code points would need to be allocated.

The first option has been chosen here. Multiple back-to-back IOAM options can be encoded as a succession of IOAM headers, with the same single GPE protocol type appearing as the next protocol before each IOAM header, but different sub-types within each IOAM header.

#### 4.2. IOAM and the Use of the VXLAN-GPE O-bit

[I-D.ietf-nvo3-vxlan-gpe] defines an "O bit" for OAM packets. Per [I-D.ietf-nvo3-vxlan-gpe] the O bit indicates that the packet contains an OAM message instead of data payload. Packets that carry IOAM data fields in addition to regular data payload / customer traffic must not set the O bit. Packets that carry only IOAM data fields without any payload must set the O bit.

#### 4.3. Transit Devices

If IOAM is deployed in domains where UDP port numbers are not controlled and do not have a domain-wide meaning, such as on the global Internet, transit devices MUST NOT attempt to modify the IOAM data contained in the IOAM header following the VXLAN-GPE header. In case UDP port numbers are not controlled there might be UDP packets specifying the same UDP port number that VXLAN-GPE utilizes, i.e. 4790, but with a payload that is not VXLAN-GPE. The scenario and associated reasoning is discussed in [RFC7605] which states that "it is important to recognize that any interpretation of port numbers -- except at the endpoints -- may be incorrect because port numbers are meaningful only at the endpoints."

### 5. IANA Considerations

#### 5.1. VXLAN-GPE Next Protocol Value

IANA is requested to allocate a value in the VXLAN-GPE "Next Protocol" registry for IOAM, which is defined in [I-D.ietf-nvo3-vxlan-gpe].

Next Protocol	Description	Reference
0x81	IOAM	This document

## 5.2. LISP-GPE Next Protocol Value

IANA is requested to allocate a value in the LISP-GPE "Next Protocol" registry for IOAM, which is defined in [RFC9305].

Next Protocol	Description	Reference
0x81	IOAM	This document

## 6. Security Considerations

The security considerations of VXLAN-GPE are discussed in [I-D.ietf-nvo3-vxlan-gpe], and the security considerations of IOAM in general are discussed in [RFC9197].

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

[I-D.ietf-nvo3-vxlan-gpe]  
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN (VXLAN-GPE)", Work in Progress, Internet-Draft, draft-ietf-nvo3-vxlan-gpe-13, 4 November 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-nvo3-vxlan-gpe-13>>.

- [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>>.
- [RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <<https://www.rfc-editor.org/info/rfc7605>>.
- [RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <<https://www.rfc-editor.org/info/rfc9197>>.
- [RFC9305] Maino, F., Ed., Lemon, J., Agarwal, P., Lewis, D., and M. Smith, "Locator/ID Separation Protocol (LISP) Generic Protocol Extension", RFC 9305, DOI 10.17487/RFC9305, October 2022, <<https://www.rfc-editor.org/info/rfc9305>>.

## 8.2. Informative References

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

### Authors' Addresses

Frank Brockners (editor)  
Cisco Systems, Inc.  
Hansaallee 249, 3rd Floor  
40549 DUESSELDORF  
Germany  
Email: [fbrockne@cisco.com](mailto:fbrockne@cisco.com)

Shwetha Bhandari  
Thoughtspot  
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout  
Bangalore, KARNATAKA 560 102  
India  
Email: [shwetha.bhandari@thoughtspot.com](mailto:shwetha.bhandari@thoughtspot.com)

Vengada Prasad Govindan  
Cisco Systems, Inc.  
Email: [venggovi@cisco.com](mailto:venggovi@cisco.com)

Carlos Pignataro (editor)  
North Carolina State University  
United States of America  
Email: cpignata@gmail.com, cmpignat@ncsu.edu

Hannes Gredler  
RtBrick Inc.  
Email: hannes@rtbrick.com

John Leddy  
Email: john@leddy.net

Stephen Youell  
JP Morgan Chase  
25 Bank Street  
London  
E14 5JP  
United Kingdom  
Email: stephen.youell@jpmorgan.com

Tal Mizrahi (editor)  
Huawei  
8-2 Matam  
Haifa 3190501  
Israel  
Email: tal.mizrahi.phd@gmail.com

Aviv Kfir  
Nvidia  
350 Oakmead Parkway, Suite 100  
Sunnyvale, CA, 94085  
United States of America  
Email: avivk@nvidia.com

Barak Gafni  
Nvidia  
350 Oakmead Parkway, Suite 100  
Sunnyvale, CA, 94085  
United States of America  
Email: gbarak@nvidia.com

Petr Lapukhov  
Facebook  
1 Hacker Way  
Menlo Park, CA, 94025  
United States of America  
Email: petr@fb.com

Mickey Spiegel  
Barefoot Networks, an Intel company  
4750 Patrick Henry Drive  
Santa Clara, CA, 95054  
United States of America  
Email: mickey.spiegel@intel.com