Geneve encapsulation for In-situ OAM Data
draft-brockners-ippm-ioam-geneve-03

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 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."
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
- OAM: Operations, Administration, and Maintenance
- Geneve: Generic Network Virtualization Encapsulation

3. IOAM Data Field Encapsulation in Geneve

Geneve is defined in [I-D.ietf-nvo3-geneve]. IOAM data fields are carried in the Geneve header as a tunnel option, using a single Geneve Option Class TBD_IOAM. The different IOAM data fields defined in [I-D.ietf-ippm-ioam-data] are added as TLVs using that Geneve Option Class. In an administrative domain where IOAM is used, insertion of the IOAM header in Geneve is enabled at the Geneve tunnel endpoints, which also serve as IOAM encapsulating/decapsulating nodes by means of configuration.
The Geneve header and fields are defined in [I-D.ietf-nvo3-geneve]. The Geneve Option Class value for use with IOAM is TBD_IOAM.

The fields related to the encapsulation of IOAM data fields in Geneve are defined 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 field defining the IOAM Option type, as defined in Section 7.2 of [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.

IOAM Option and Data Space: IOAM option header and data is present as defined by the Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].
Multiple IOAM options MAY be included within the Geneve encapsulation. For example, if a Geneve encapsulation contains two IOAM options before a data payload, there would be two fields with TBD_IOAM Option Class each, differentiated by the Type field which specifies the type of the IOAM data included.

4. Considerations

This section summarizes a set of considerations on the overall approach taken for IOAM data encapsulation in Geneve, 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 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 \((2^5 \times 4)\) 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.

4.2. IOAM and the use of the Geneve O-bit

[I-D.ietf-nvo3-geneve] defines an "O bit" for OAM packets. Per [I-D.ietf-nvo3-geneve] the O bit indicates that the packet contains a control 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 option class. In case UDP port numbers are not controlled there might be UDP packets, which leverage the UDP port number that Geneve utilizes, i.e. 6081, but the payload of these packets isn’t Geneve. 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

IANA is requested to allocate a Geneve "option class" numbers for IOAM:
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. Normative References

[I-D.ietf-ippm-ioam-data]

[I-D.ietf-nvo3-geneve]


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  
Huawei Network.IO Innovation Lab  
Israel  
Email: tal.mizrahi.phd@gmail.com  

Petr Lapukhov  
Facebook  
1 Hacker Way  
Menlo Park, CA  94025  
US  
Email: petr@fb.com  

Barak Gafni  
Mellanox Technologies, Inc.  
350 Oakmead Parkway, Suite 100  
Sunnyvale, CA  94085  
U.S.A.  
Email: gbarak@mellanox.com