Stream: Internet Engineering Task Force (IETF)
RFC: 9452
Category: Standards Track
Published: August 2023
ISSN: 2070-1721
Authors: F. Brockners, Ed. *Cisco*
S. Bhandari, Ed. *Thoughtspot*

RFC 9452
Network Service Header (NSH) Encapsulation for In Situ OAM (IOAM) Data

Abstract

In situ Operations, Administration, and Maintenance (IOAM) is used for recording and collecting operational and telemetry information while the packet traverses a path between two points in the network. This document outlines how IOAM-Data-Fields are encapsulated with the Network Service Header (NSH).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9452.

Copyright Notice

Table of Contents

1. Introduction

IOAM, as defined in [RFC9197], is used to record and collect OAM information 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 what 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) encapsulation [RFC8300] for the Service Function Chaining (SFC) Architecture [RFC7665]. The IOAM-Data-Fields are defined in [RFC9197].

2. Conventions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Abbreviations used in this document:

IOAM: In situ Operations, Administration, and Maintenance

MD: NSH Metadata, see [RFC7665]

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

SFC: Service Function Chaining

TLV: Type, Length, Value

3. IOAM Encapsulation with NSH

The NSH is defined in [RFC8300]. IOAM-Data-Fields are carried as NSH payload using a Next Protocol header that follows the NSH headers. An IOAM header containing the IOAM-Data-Fields is added. The IOAM-Data-Fields **MUST** follow the definitions corresponding to IOAM Option-Types (e.g., see Section 4 of [RFC9197] and Section 3.2 of [RFC9326]). In an administrative domain where IOAM is used, insertion of the IOAM header in NSH is enabled at the NSH tunnel endpoints, which are also configured to serve as encapsulating and decapsulating nodes for IOAM. The operator **MUST** ensure that SFC-aware nodes along the Service Function Path support IOAM; otherwise, packets might be dropped (see the last paragraph of this section as well as Section 2.2 of [RFC8300]). The IOAM transit nodes (e.g., a Service Function Forwarder (SFF)) **MUST** process all the IOAM headers that are relevant based on its configuration. See [RFC9378] for a discussion of deployment-related aspects of IOAM-Data-Fields.

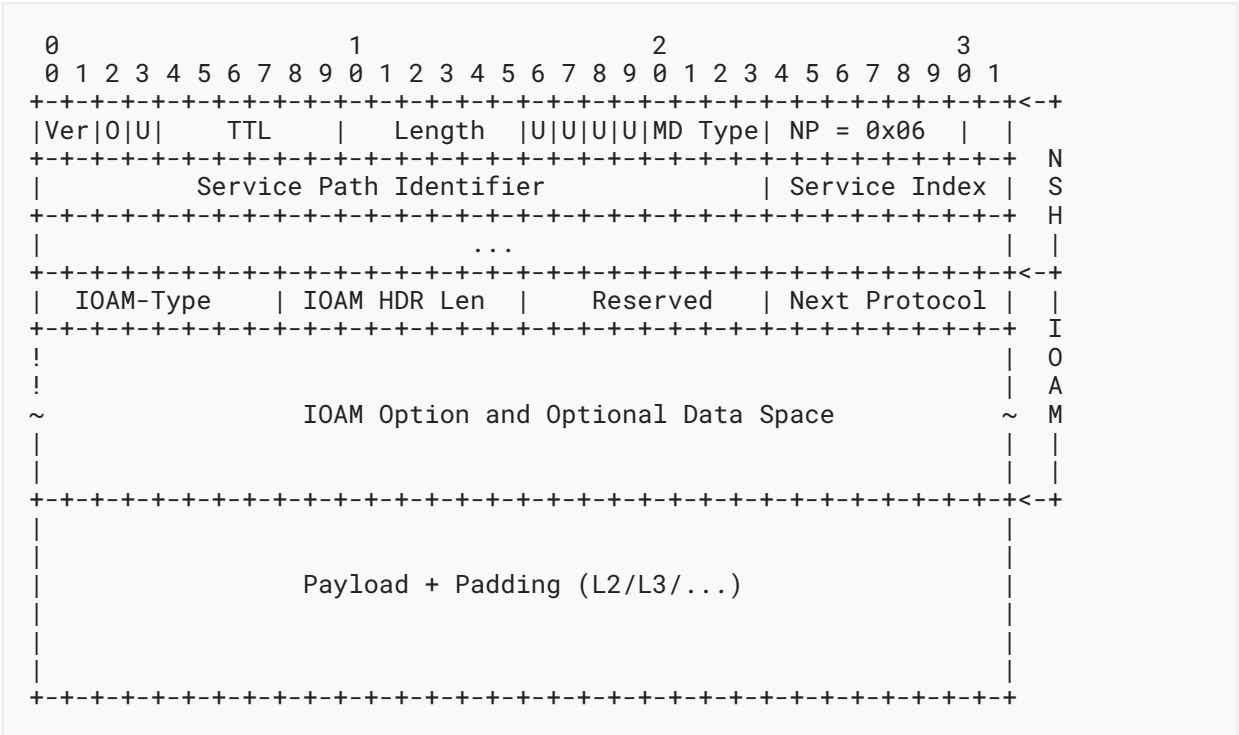

Figure 1

The NSH header and fields are defined in [RFC8300]. The O bit **MUST** be handled following the rules in [RFC9451]. The "NSH Next Protocol" value (referred to as "NP" in the diagram above) is 0x06.

The IOAM-related fields in NSH are defined as follows:

IOAM-Type:
: 8-bit field defining the IOAM Option-Type, as defined in the "IOAM Option-Type" registry specified in [RFC9197].

IOAM HDR Len:
: 8-bit field that contains the length of the IOAM header in multiples of 4-octets, including the "IOAM-Type" and "IOAM HDR Len" fields.

Reserved bits:
: Reserved bits are present for future use. The reserved bits **MUST** be set to 0x0 upon transmission and ignored upon receipt.

Next Protocol:
: 8-bit unsigned integer that determines the type of header following IOAM. The semantics of this field are identical to the Next Protocol field in [RFC8300].

IOAM Option and Optional Data Space:
: IOAM-Data-Fields as specified by the IOAM-Type field. IOAM-Data-Fields are defined corresponding to the IOAM Option-Type (e.g., see Section 4 of [RFC9197] and Section 3.2 of [RFC9326]) and are always aligned by 4 octets. Thus, there is no padding field.

Multiple IOAM Option-Types **MAY** be included within the NSH encapsulation. For example, if an NSH encapsulation contains two IOAM Option-Types before a data payload, the Next Protocol field of the first IOAM option will contain the value 0x06, while the Next Protocol field of the second IOAM Option-Type will contain the "NSH Next Protocol" number indicating the type of the data payload. The applicability of the IOAM Active and Loopback flags [RFC9322] is outside the scope of this document and may be specified in the future.

In case the IOAM Incremental Trace Option-Type is used, an SFC-aware node that serves as an IOAM transit node needs to adjust the "IOAM HDR Len" field accordingly. See Section 4.4 of [RFC9197].

Per Section 2.2 of [RFC8300], packets with unsupported Next Protocol values **SHOULD** be silently dropped by default. Thus, when a packet with IOAM is received at an NSH-based forwarding node (such as an SFF) that does not support the IOAM header, it **SHOULD** drop the packet. The mechanisms to maintain and notify of such events are outside the scope of this document.

4. IANA Considerations

IANA has allocated the following code point for IOAM in the "NSH Next Protocol" registry:

Next Protocol	Description	Reference
0x06	IOAM (Next Protocol is an IOAM header)	RFC 9452

Table 1

5. Security Considerations

IOAM is considered a "per domain" feature, where the operator decides how to leverage and configure IOAM according to the operator's needs. The operator needs to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain. For additional IOAM-related security considerations, see Section 9 of [RFC9197]. For additional OAM- and NSH-related security considerations, see Section 5 of [RFC9451].

6. References

6.1. Normative References

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[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>.

[RFC9451] Boucadair, M., "Operations, Administration, and Maintenance (OAM) Packet and Behavior in the Network Service Header (NSH)", RFC 9451, DOI 10.17487/RFC9451, August 2023, <https://www.rfc-editor.org/info/rfc9451>.

6.2. Informative References

[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>.

[RFC9322] Mizrahi, T., Brockners, F., Bhandari, S., Gafni, B., and M. Spiegel, "In Situ Operations, Administration, and Maintenance (IOAM) Loopback and Active Flags", RFC 9322, DOI 10.17487/RFC9322, November 2022, <https://www.rfc-editor.org/info/rfc9322>.

[RFC9326] Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", RFC 9326, DOI 10.17487/RFC9326, November 2022, <https://www.rfc-editor.org/info/rfc9326>.

[RFC9378] Brockners, F., Ed., Bhandari, S., Ed., Bernier, D., and T. Mizrahi, Ed., "In Situ Operations, Administration, and Maintenance (IOAM) Deployment", RFC 9378, DOI 10.17487/RFC9378, April 2023, <https://www.rfc-editor.org/info/rfc9378>.

Appendix A. Discussion of the IOAM-Encapsulation Approach

This section lists several approaches considered for encapsulating IOAM with NSH and presents the rationale for the approach chosen in this document.

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 lookups of fields within the packet. Any operation that 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 should be kept to a minimum. This means that flat TLV structures are preferred over nested TLV structures. IOAM-Data-Fields are grouped into several categories, including trace, proof-of-transit, and edge-to-edge. Each of these options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure and would instead carry the options as a serial sequence.

- Total length of the IOAM-Data-Fields:

  The total length of IOAM-Data-Fields can grow quite large if multiple different IOAM-Data-Fields are used and large path-lengths need to be considered. For example, if an operator would consider using the IOAM Trace Option-Type and capture node-id, app_data, egress and ingress interface-id, timestamp seconds, and timestamp nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In this case, the particular deployment has a maximum path length of 15 hops in the IOAM domain, and a maximum of 300 octets would be encapsulated in the packet.

Different approaches for encapsulating IOAM-Data-Fields in NSH could be considered:

1. Encapsulation of IOAM-Data-Fields as "NSH MD Type 2" (see [RFC8300], Section 2.5).

Each IOAM Option-Type (e.g., 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 metadata. 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 Option-Type (e.g., trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol".

3. Encapsulation of IOAM-Data-Fields using the "Next Protocol" field.

   A single NSH protocol type code point would be allocated for IOAM. A "sub-type" field would then specify what IOAM options type (trace, proof-of-transit, edge-to-edge) is carried.

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

Acknowledgments

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

Contributors

The following people contributed significantly to the content of this document and should be considered coauthors:

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 of America
Email: cpignata@cisco.com

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

David Mozes
Email: mosesster@gmail.com

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

Remy Chang
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
United States of America

Authors' Addresses

Frank Brockners (EDITOR)
Cisco Systems, Inc.
3rd Floor
Hansaallee 249
40549 Duesseldorf
Germany
Email: fbrockne@cisco.com

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