Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 17, 2022

T. Mizrahi
Huawei
J. Iurman
ULiege
F. Brockners
Cisco

January 13, 2022

In Situ OAM Profile for the Linux Kernel Implementation
draft-mizrahi-ippm-ioam-linux-profile-02

Abstract

In Situ Operations, Administration and Maintenance (IOAM) is used for monitoring network performance and for detecting traffic bottlenecks and anomalies. This document defines an IOAM profile that is used in the Linux kernel implementation, starting from the Linux 5.15 kernel.

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 July 17, 2022.

Table of Contents

1. Introduction

IOAM [I-D.ietf-ippm-ioam-data] is used for monitoring traffic in the network by incorporating IOAM data fields into in-flight data packets.

An IOAM profile [I-D.mizrahi-ippm-ioam-profile] defines a use case or a set of use cases for IOAM, and an associated set of rules that restrict the scope and features of the IOAM specification, thereby limiting it to a subset of the full functionality.

This document introduces a profile of IOAM that is used in the Linux kernel implementation. The profile is intended to formally specify the subset of features that are in scope, and to enable other implementations to interoperate with the Linux implementation.

2. The Linux IOAM Profile

2.1. Use Cases

The Linux kernel implementation enables the functionality of any of the following nodes:

- IOAM encapsulating node
- IOAM transit node

- IOAM decapsulating node

One possible use case is a set of Linux-based hosts that function as IOAM encapsulating and decapsulating nodes, interconnected by IOAM transit nodes that are not necessarily Linux-based. Thus, Linux-based implementations are expected to interoperate with other implementations that comply to this profile.

Another possible use case is a homogenous setting in which all IOAM nodes are Linux-based.

2.2. IOAM Version

The current profile is based on [I-D.ietf-ippm-ioam-data-15], which is a work-in-progress version of IOAM.

2.3. IOAM Options

The current profile uses the Pre-allocated Trace Option-Type. It is assumed that one IOAM option is used in an IOAM encapsulated packet.

2.4. Encapsulation

This profile uses an IPv6 encapsulation for the IOAM option, i.e., the option is encapsulated in an IPv6 Extension Header. Generally speaking, this extension header may be an extension of an IPv6 tunnel header, or it may be an extension of the end-to-end IPv6 header. Both cases are discussed below. The extension header is used for the IOAM Pre-allocated Trace Option-Type, as defined in [I-D.ietf-ippm-ioam-ipv6-options-06], which is a work-in-progress version of the IPv6 IOAM option.

The IPv6 Extension Header is a Hop-by-Hop Options header, that contains the IOAM Trace Option-Type. The Hop-by-Hop Options header can include one or more options, such that one of these options is the IOAM Pre-allocated Trace Option-Type. Figure 1 illustrates the format of this Hop-by-Hop Options header when the IOAM Pre-allocated Trace Option-Type is the only Hop-by-Hop option. If more options are present the format will change accordingly.

As illustrated in Figure 1, the first 2 octets are the Hop-by-Hop Options header [RFC8200], followed by a 2 octet Padding field. The following 4 octets are the IOAM IPv6 option header [I-D.ietf-ippm-ioam-ipv6-options-06]. The IOAM Option includes the 8 octet Pre-allocated Trace Option-Type header [I-D.ietf-ippm-ioam-data-15], followed by the Option Data.

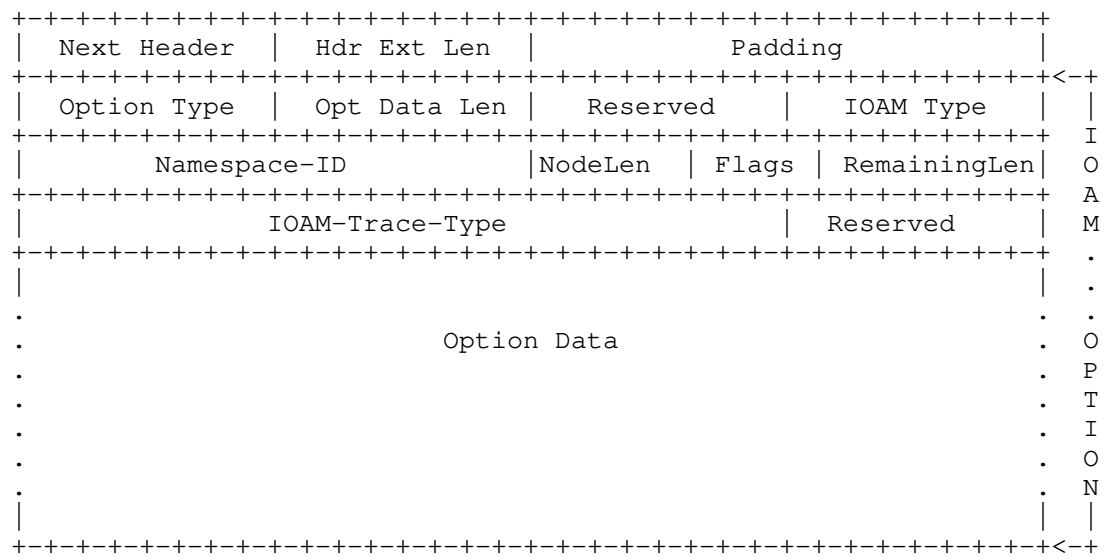

Figure 1: IPv6 IOAM Extension Header Format

Starting from the Linux 5.16 kernel, the IOAM encapsulation might also use an IPv6 tunnel for in-transit packets, as defined in [RFC2473], and as illustrated in Figure 2. The kernel supports both alternatives: either encapsulation using an IPv6 tunnel, or pushing the IOAM option as an IPv6 extension header of the existing IPv6 header.

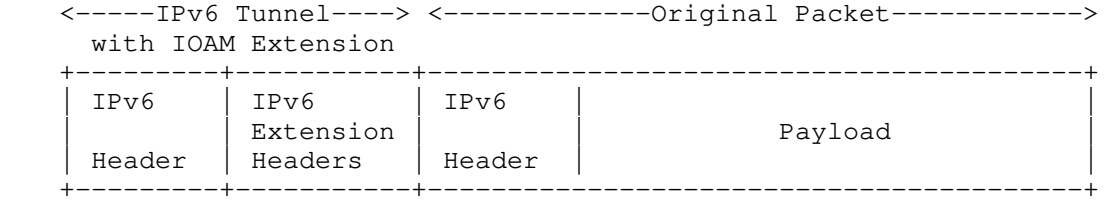

Figure 2: IOAM in IPv6 Tunnel Encapsulation

2.5. IOAM Supported Data Fields

The current profile supports all the data field types that are defined in [I-D.ietf-ippm-ioam-data-15] for the Pre-allocated Trace Option-Type, except for the Checksum Complement field, which is not required in this profile, since the IOAM Trace Option is encapsulated directly in an IPv6 Extension Header, without any additional layers that use a checksum.

2.6. Trace Option-Type Flags

This profile only uses the Overflow flag.

2.7. Timestamp Format

This profile uses the POSIX timestamp format.

2.8. Profile Coexistence

It is assumed that the current profile is used in a confined administrative domain in which no other IOAM profiles are used. Therefore, it is assumed that the current profile does not coexist with other profiles.

2.9. Validity

An IOAM transit/decapsulating node that receives a packet with IOAM options that do not comply to the current profile is expected to forward/decapsulate the packet without IOAM processing, if it is able to do so. If a decapsulating node is not able to decapsulate an IOAM option that is not compliant to the current profile, the packet is discarded.

3. Notes about the IOAM Support in Linux

The current Linux implementation supports all the data field types defined in [I-D.ietf-ippm-ioam-data-15] for the Pre-allocated Trace Option-Type. Specifically, the Linux implementation does not update the transit delay, the queue depth, the checksum complement and the buffer occupancy. These four data field types are passively supported, meaning the Linux implementation can add the Pre-allocated Trace Option-Type including these fields, but cannot populate them with system information. They are populated with empty values and, therefore, interoperability is possible with other IOAM nodes that support these fields.

The following table summarizes the data field type support in the Linux implementation.

Data field type	Status
Hop_Lim and node_id (short format)	Supported
Ingress_if_id and egress_if_id (short format)	Supported
Timestamp seconds	Supported
Timestamp fraction	Supported
Transit delay	Passive support
Namespace specific data (short format)	Supported
Queue depth	Supported*
Checksum complement	Passive support
Hop_Lim and node_id (wide format)	Supported
Ingress_if_id and egress_if_id (wide format)	Supported
Namespace specific data (wide format)	Supported
Buffer occupancy	Passive support
Opaque State Snapshot	Supported

*: Queue depth is supported starting from the Linux 5.17 kernel, and has passive support for older kernel versions.

Both the Opaque State Snapshot and the Namespace specific data are supported in the Linux implementation by incorporating configurable values into these fields. Notably, Linux-based IOAM nodes can interoperate with other nodes that use the Opaque State Snapshot and/or the Namespace specific data in a more flexible way.

If an IOAM transit node receives a packet with one or more undefined bits of the trace type set to 1, it will add corresponding node data filled with the reserved value 0xFFFFFFFF, as defined in [I-D.ietf-ippm-ioam-data-15]. This allows for some flexibility from an interoperability point of view.

4. IANA Considerations

This document does not include any requests from IANA.

5. Security Considerations

The security considerations of IOAM profiles are discussed in [I-D.mizrahi-ippm-ioam-profile]. The current document does not present any new security considerations.

6. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-17 (work in progress), December 2021.

[I-D.ietf-ippm-ioam-data-15]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-15 (work in progress), October 2021.

[I-D.ietf-ippm-ioam-ipv6-options-06]
Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-06 (work in progress), July 2021.

[I-D.mizrahi-ippm-ioam-profile]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., Zhou, T., and J. Lemon, "In Situ OAM Profiles", draft-mizrahi-ippm-ioam-profile-05 (work in progress), August 2021.

[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>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

Authors' Addresses

Tal Mizrahi
Huawei
8-2 Matam
Haifa
Israel

Email: tal.mizrahi.phd@gmail.com

Justin Iurman
Universite de Liege
10, Allee de la decouverte (B28)
Sart-Tilman, LIEGE 4000
Belgium

Email: justin.iurman@uliege.be

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

Email: fbrockne@cisco.com