                          In-situ OAM IPv6 Options
                    draft-ietf-ippm-ioam-ipv6-options-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 IPv6.

Status of This Memo

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 August 25, 2021.

Copyright Notice

Table of Contents

1. Introduction

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 the IPv6 [RFC8200]
and discusses deployment options for networks that use
IPv6-encapsulated IOAM data fields. These options have distinct
deployment considerations; for example, the IOAM domain can either be
between hosts, or be between IOAM encapsulating and decapsulating
network nodes that forward traffic, such as routers.

2. Conventions

2.1. Requirements Language

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.

2.2. Abbreviations

Abbreviations used in this document:

E2E: Edge-to-Edge

IOAM: In-situ Operations, Administration, and Maintenance

ION: IOAM Overlay Network

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

3. In-situ OAM Metadata Transport in IPv6

In-situ OAM in IPv6 is used to enhance diagnostics of IPv6 networks.
It complements other mechanisms designed to enhance diagnostics of
IPv6 networks, such as the IPv6 Performance and Diagnostic Metrics
Destination Option described in [RFC8250].

IOAM data fields can be encapsulated in "option data" fields using
two types of extension headers in IPv6 packets - either Hop-by-Hop
Options header or Destination options header. Deployments select one
of these extension header types depending on how IOAM is used, as
described in section 4 of [I-D.ietf-ippm-ioam-data]. Multiple

options with the same Option Type MAY appear in the same Hop-by-Hop
Options or Destination Options header, with distinct content.

In order for IOAM to work in IPv6 networks, IOAM MUST be explicitly
enabled per interface on every node within the IOAM domain. Unless a
particular interface is explicitly enabled (i.e., explicitly
configured) for IOAM, a router MUST drop packets that contain
extension headers carrying IOAM data-fields. This is the default
behavior and is independent of whether the Hop-by-Hop options or
Destination options are used to encode the IOAM data. This ensures
that IOAM data does not unintentionally get forwarded outside the
IOAM domain.

An IPv6 packet carrying IOAM data in an Extension header can have
other extension headers, compliant with [RFC8200].

IPv6 Hop-by-Hop and Destination Option format for carrying in-situ
OAM data fields:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  |  Opt Data Len |    Reserved   |   IOAM Type   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|                                                               |  |
.                                                               .  I
.                                                               .  O
.                                                               .  A
.                                                               .  M
.                                                               .  .
.                         Option Data                          .  O
.                                                               .  P
.                                                               .  T
.                                                               .  I
.                                                               .  O
.                                                               .  N
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
```

Option Type: 8-bit option type identifier as defined in Section 6.

Opt Data Len: 8-bit unsigned integer. Length of this option, in
   octets, not including the first 2 octets.

Reserved: 8-bit field MUST be set to zero upon transmission and
   ignored upon reception.

IOAM Type: 8-bit field as defined in section 7.2 in
   [I-D.ietf-ippm-ioam-data].

Option Data: Variable-length field. Option-Type-specific data.

In-situ OAM Option-Types are inserted as Option data as follows:

1. Pre-allocated Trace Option: The in-situ OAM Preallocated Trace
    Option-Type defined in [I-D.ietf-ippm-ioam-data] is represented
    as an IPv6 option in the Hop-by-Hop extension header:

    Option Type: 001xxxxx 8-bit identifier of the IOAM type of
       option. xxxxx=TBD.

    IOAM Option-Type: IOAM Pre-allocated Trace Option-Type.

2. Incremental Trace Option: The in-situ OAM Incremental Trace
    Option-Type defined in [I-D.ietf-ippm-ioam-data] is represented
    as an IPv6 option in the Hop-by-Hop extension header:

    Option Type: 001xxxxx 8-bit identifier of the IOAM type of
       option. xxxxx=TBD.

    IOAM Option-Type: IOAM Incremental Trace Option-Type.

3. Proof of Transit Option: The in-situ OAM POT Option-Type defined
    in [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in
    the Hop-by-Hop extension header:

    Option Type: 001xxxxx 8-bit identifier of the IOAM type of
       option. xxxxx=TBD.

    IOAM Option-Type: IOAM POT Option-Type.

4. Edge to Edge Option: The in-situ OAM E2E option defined in
    [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in
    Destination extension header:

    Option Type: 000xxxxx 8-bit identifier of the IOAM type of
       option. xxxxx=TBD.

    IOAM Option-Type: IOAM E2E Option-Type.

5. Direct Export (DEX) Option: The in-situ OAM Direct Export Option-
    Type defined in [I-D.ietf-ippm-ioam-direct-export] is represented
    as an IPv6 option in the Hop-by-Hop extension header:

Option Type: 000xxxxx 8-bit identifier of the IOAM type of
    option. xxxxx=TBD.

IOAM Option-Type: IOAM Direct Export (DEX) Option-Type.

All the in-situ OAM IPv6 options defined here have alignment
requirements. Specifically, they all require 4n alignment. This
ensures that fields specified in [I-D.ietf-ippm-ioam-data] are
aligned at a multiple-of-4 offset from the start of the Hop-by-Hop
and Destination Options header. In addition, to maintain IPv6
extension header 8-octet alignment and avoid the need to add or
remove padding at every hop, the Trace-Type for Incremental Trace
Option in IPv6 MUST be selected such that the IOAM node data length
is a multiple of 8-octets.

IPv6 options can have a maximum length of 255 octets. Consequently,
the total lenght of IOAM Option-Types including all data fields is
also limited to 255 octets when encapsulated into IPv6.

4. IOAM Deployment In IPv6 Networks

4.1. Considerations for IOAM deployment in IPv6 networks

IOAM deployments in IPv6 networks should take the following
considerations and requirements into account:

C1 It is desirable that the addition of IOAM data fields neither
   changes the way routers forward packets nor the forwarding
   decisions the routers take. Packets with added OAM information
   should follow the same path within the domain that an identical
   packet without OAM information would follow, even in the presence
   of ECMP. Such behavior is particularly important for deployments
   where IOAM data fields are only added "on-demand", e.g., to
   provide further insights in case of undesired network behavior for
   certain flows. Implementations of IOAM SHOULD ensure that ECMP
   behavior for packets with and without IOAM data fields is the
   same.

C2 Given that IOAM data fields increase the total size of a packet,
   the size of a packet including the IOAM data could exceed the
   PMTU. In particular, the incremental trace IOAM Hop-by-Hop (HbH)
   Option, which is intended to support hardware implementations of
   IOAM, changes Option Data Length en-route. Operators of an IOAM
   domain SHOULD ensure that the addition of OAM information does not
   lead to fragmentation of the packet, e.g., by configuring the MTU
   of transit routers and switches to a sufficiently high value.
   Careful control of the MTU in a network is one of the reasons why
   IOAM is considered a domain-specific feature (see also

[I-D.ietf-ippm-ioam-data]). In addition, the PMTU tolerance range
in the IOAM domain should be identified (e.g., through
configuration) and IOAM encapsulation operations and/or IOAM data
field insertion (in case of incremental tracing) should not be
performed if it exceeds the packet size beyond PMTU.

C3 Packets with IOAM data or associated ICMP errors, should not
arrive at destinations that have no knowledge of IOAM. For
exmample, if IOAM is used in in transit devices, misleading ICMP
errors due to addition and/or presence of OAM data in a packet
could confuse the host that sent the packet if it did not insert
the OAM information.

C4 OAM data leaks can affect the forwarding behavior and state of
network elements outside an IOAM domain. IOAM domains SHOULD
provide a mechanism to prevent data leaks or be able to ensure
that if a leak occurs, network elements outside the domain are not
affected (i.e., they continue to process other valid packets).

C5 The source that inserts and leaks the IOAM data needs to be easy
to identify for the purpose of troubleshooting, due to the high
complexity of troubleshooting a source that inserted the IOAM data
and did not remove it when the packet traversed across an
Autonomous System (AS). Such a troubleshooting process might
require coordination between multiple operators, complex
configuration verification, packet capture analysis, etc.

C6 Compliance with [RFC8200] requires OAM data to be encapsulated
instead of header/option insertion directly into in-flight packets
using the original IPv6 header.

4.2. IOAM domains bounded by hosts

For deployments where the IOAM domain is bounded by hosts, hosts will
perform the operation of IOAM data field encapsulation and
decapsulation. IOAM data is carried in IPv6 packets as Hop-by-Hop or
Destination options as specified in this document.

4.3. IOAM domains bounded by network devices

For deployments where the IOAM domain is bounded by network devices,
network devices such as routers form the edge of an IOAM domain.
Network devices will perform the operation of IOAM data field
encapsulation and decapsulation.

4.4. Deployment options

   This section lists out possible deployment options that can be
   employed to meet the requirements listed in Section 4.1.

4.4.1. IPv6-in-IPv6 encapsulation

   The "IPv6-in-IPv6" approach preserves the original IP packet and add
   an IPv6 header including IOAM data fields in an extension header in
   front of it, to forward traffic within and across an IOAM domain.
   The overlay network formed by the additional IPv6 header with the
   IOAM data fields included in an extension header is referred to as
   IOAM Overlay Network (ION) in this document.

   The following steps should be taken to perform an IPv6-in-IPv6
   approach:

   1. The source address of the outer IPv6 header is that of the IOAM
       encapsulating node. The destination address of the outer IPv6
       header is the same as the inner IPv6 destination address, i.e.,
       the destination address of the packet does not change.

   2. To simplify debugging in case of leaked IOAM data fields,
       consider a new IOAM E2E destination option to identify the Source
       IOAM domain (AS, v6 prefix). Insert this option into the IOAM
       destination options EH attached to the outer IPv6 header. This
       additional information would allow for easy identification of an
       AS operator that is the source of packets with leaked IOAM
       information. Note that leaked packets with IOAM data fields
       would only occur in case a router would be misconfigured.

   3. All the IOAM options are defined with type "00" - skip over this
       option and continue processing the header. Presence of these
       options must not cause packet drops in network elements that do
       not understand the option. In addition,
       [I-D.ietf-6man-hbh-header-handling] should be considered.

4.4.2. IP-in-IPv6 encapsulation with ULA

   The "IP-in-IPv6 encapsulation with ULA" [RFC4193] approach can be
   used to apply IOAM to either an IPv6 or an IPv4 network. In
   addition, it fulfills requirement C4 (avoid leaks) by using ULA for
   the ION. Similar to the IPv6-in-IPv6 encapsulation approach above,
   the original IP packet is preserved. An IPv6 header including IOAM
   data fields in an extension header is added in front of it, to
   forward traffic within and across the IOAM domain. IPv6 addresses
   for the ION, i.e. the outer IPv6 addresses are assigned from the ULA
   space. Addressing and routing in the ION are to be configured so

that the IP-in-IPv6 encapsulated packets follow the same path as the
original, non-encapsulated packet would have taken. This would
create an internal IPv6 forwarding topology using the IOAM domain's
interior ULA address space which is parallel with the forwarding
topology that exists with the non-IOAM address space (the topology
and address space that would be followed by packets that do not have
supplemental IOAM information). Establishment and maintenance of the
parallel IOAM ULA forwarding topology could be automated, e.g.,
similar to how LDP [RFC5036] is used in MPLS to establish and
maintain an LSP forwarding topology that is parallel to the network's
IGP forwarding topology.

Transit across the ION could leverage the transit approach for
traffic between BGP border routers, as described in [RFC1772], "A.2.3
Encapsulation". Assuming that the operational guidelines specified
in Section 4 of [RFC4193] are properly followed, the probability of
leaks in this approach will be almost close to zero. If the packets
do leak through IOAM egress device misconfiguration or partial IOAM
egress device failure, the packets' ULA destination address is
invalid outside of the IOAM domain. There is no exterior destination
to be reached, and the packets will be dropped when they encounter
either a router external to the IOAM domain that has a packet filter
that drops packets with ULA destinations, or a router that does not
have a default route.

4.4.3. x-in-IPv6 Encapsulation that is used Independently

In some cases it is desirable to monitor a domain that uses an
overlay network that is deployed independently of the need for IOAM,
e.g., an overlay network that runs Geneve-in-IPv6, or VXLAN-in-IPv6.
In this case IOAM can be encapsulated in as an extension header in
the tunnel (outer) IPv6 header. Thus, the tunnel encapsulating node
is also the IOAM encapsulating node, and the tunnel end point is also
the IOAM decapsulating node.

5. Security Considerations

This document describes the encapsulation of IOAM data fields in
IPv6. Security considerations of the specific IOAM data fields for
each case (i.e., Trace, Proof of Transit, and E2E) are described and
defined in [I-D.ietf-ippm-ioam-data].

As this document describes new options for IPv6, these are similar to
the security considerations of [RFC8200] and the weakness documented
in [RFC8250].

6. IANA Considerations

This draft requests the following IPv6 Option Type assignments from
the Destination Options and Hop-by-Hop Options sub-registry of
Internet Protocol Version 6 (IPv6) Parameters.

http://www.iana.org/assignments/ipv6-parameters/ipv6-
parameters.xhtml#ipv6-parameters-2

Hex Value	Binary Value act chg rest			Description	Reference
TBD_1_0	00	0	TBD_1	IOAM	[This draft]
TBD_1_1	00	1	TBD_1	IOAM	[This draft]

7. Acknowledgements

The authors would like to thank Tom Herbert, Eric Vyncke, Nalini
Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra
Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik
Nordmark, LJ Wobker, Mark Smith, Andrew Yourtchenko and Justin Iurman
for the comments and advice. For the IPv6 encapsulation, this
document leverages concepts described in
[I-D.kitamura-ipv6-record-route]. The authors would like to
acknowledge the work done by the author Hiroshi Kitamura and people
involved in writing it.

8. References

8.1. 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-11 (work in
          progress), November 2020.

[I-D.ietf-ippm-ioam-direct-export]
          Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F.,
          Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ
          OAM Direct Exporting", draft-ietf-ippm-ioam-direct-
          export-02 (work in progress), November 2020.

[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>.

8.2. Informative References

   [I-D.ietf-6man-hbh-header-handling]
               Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options
               Extension Header", March 2016.

   [I-D.kitamura-ipv6-record-route]
               Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop
               Option Extension", draft-kitamura-ipv6-record-route-00
               (work in progress), November 2000.

   [RFC1772]   Rekhter, Y. and P. Gross, "Application of the Border
               Gateway Protocol in the Internet", RFC 1772,
               DOI 10.17487/RFC1772, March 1995,
               <https://www.rfc-editor.org/info/rfc1772>.

   [RFC4193]   Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast
               Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005,
               <https://www.rfc-editor.org/info/rfc4193>.

   [RFC5036]   Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed.,
               "LDP Specification", RFC 5036, DOI 10.17487/RFC5036,
               October 2007, <https://www.rfc-editor.org/info/rfc5036>.

   [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>.

   [RFC8250]   Elkins, N., Hamilton, R., and M. Ackermann, "IPv6
               Performance and Diagnostic Metrics (PDM) Destination
               Option", RFC 8250, DOI 10.17487/RFC8250, September 2017,
               <https://www.rfc-editor.org/info/rfc8250>.

Authors' Addresses

   Shwetha Bhandari
   Thoughtspot
   3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
   Bangalore, KARNATAKA 560 102
   India

   Email: shwetha.bhandari@thoughtspot.com

Frank Brockners
Cisco Systems, Inc.
Kaiserswerther Str. 115,
RATINGEN, NORDRHEIN-WESTFALEN 40880
Germany

Email: fbrockne@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

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mickey.spiegel@intel.com

Suresh Krishnan
Kaloom

Email: suresh@kaloom.com

Rajiv Asati
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: rajiva@cisco.com

Mark Smith
PO BOX 521
HEIDELBERG, VIC 3084
AU

Email: markzzzsmith+id@gmail.com