Stream: Internet Engineering Task Force (IETF)
RFC: 9322
Category: Standards Track
Published: November 2022
ISSN: 2070-1721

Authors:
T. Mizrahi, *Huawei*
F. Brockners, *Cisco*
S. Bhandari, *Thoughtspot*
B. Gafni, *Nvidia*
M. Spiegel, *Barefoot Networks*

RFC 9322
In Situ Operations, Administration, and Maintenance (IOAM) Loopback and Active Flags

Abstract

In situ Operations, Administration, and Maintenance (IOAM) collects operational and telemetry information in packets while they traverse a path between two points in the network. This document defines two new flags in the IOAM Trace Option headers, specifically the Loopback and Active flags.

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/rfc9322.

Copyright Notice

Table of Contents

1. Introduction

IOAM [RFC9197] is used for monitoring traffic in the network by incorporating IOAM data fields into in-flight data packets.

IOAM data may be represented in one of four possible IOAM options: Pre-allocated Trace, Incremental Trace, Proof of Transit (POT), and Edge-to-Edge. This document defines two new flags in the Pre-allocated and Incremental Trace options: the Loopback and Active flags.

The Loopback flag is used to request that each transit device along the path loops back a truncated copy of the data packet to the sender. The Active flag indicates that a packet is used for active measurement. The term "active measurement" in the context of this document is as defined in [RFC7799].

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. Terminology

Abbreviations used in this document:

IOAM:	In situ Operations, Administration, and Maintenance
OAM:	Operations, Administration, and Maintenance [RFC6291]

3. New IOAM Trace Option Flags

This document defines two new flags in the Pre-allocated and Incremental Trace options:

Bit 1 "Loopback" (L-bit): When set, the Loopback flag triggers the sending of a copy of a packet back towards the source, as further described in Section 4.

Bit 2 "Active" (A-bit): When set, the Active flag indicates that a packet is an active measurement packet rather than a data packet, where "active" is used in the sense defined in [RFC7799]. The packet may be an IOAM probe packet or a replicated data packet (the second and third use cases of Section 5).

4. Loopback in IOAM

The Loopback flag is used to request that each transit device along the path loops back a truncated copy of the data packet to the sender. Loopback allows an IOAM encapsulating node to trace the path to a given destination and to receive per-hop data about both the forward and return paths. Loopback is intended to provide an accelerated alternative to Traceroute that allows the encapsulating node to receive responses from multiple transit nodes along the path in less than one round-trip time (RTT) and by sending a single packet.

As illustrated in Figure 1, an IOAM encapsulating node can push an IOAM encapsulation that includes the Loopback flag onto some or all of the packets it forwards using one of the IOAM encapsulation types, e.g., [IOAM-NSH] or [IOAM-IPV6-OPTIONS]. The IOAM transit node and the decapsulating node both create copies of the packet and loop them back to the encapsulating node. The decapsulating node also terminates the IOAM encapsulation and then forwards the packet towards the destination. The two IOAM looped-back copies are terminated by the encapsulating node.

```
+--------+     +--------+     +--------+     +--------+     +--------+
|        |     |  IOAM  |.....|  IOAM  |.....|  IOAM  |     |        |
+--------+     +--------+     +--------+     +--------+     +--------+
| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |
+--------+     +--------+     +--------+     +--------+     +--------+
  Source      Encapsulating    Transit      Decapsulating   Destination
                   Node          Node           Node

               <------------  IOAM-Domain  ----------->

                    IOAM encap. with Loopback flag
Data packet  ------->============================>----------->
                                  |             |
                 IOAM looped back |             |
                    <=============+             |
                                IOAM looped back|
                    <===========================+
```

Figure 1: Loopback in IOAM

Loopback can be used only if a return path from transit nodes and destination nodes towards the source (encapsulating node) exists. Specifically, loopback is only applicable in encapsulations in which the identity of the encapsulating node is available in the encapsulation header. If an encapsulating node receives a looped-back packet that was not originated from the current encapsulating node, the packet is dropped.

4.1. Loopback: Encapsulating Node Functionality

The encapsulating node either generates synthetic packets with an IOAM trace option that has the Loopback flag set or sets the Loopback flag in a subset of the in-transit data packets. Loopback is used either proactively or on-demand, i.e., when a failure is detected. The encapsulating node also needs to ensure that sufficient space is available in the IOAM header for loopback operation, which includes transit nodes adding trace data on the original path and again on the return path.

An IOAM trace option that has the Loopback flag set **MUST** have the value '1' in the most significant bit of IOAM-Trace-Type and '0' in the rest of the bits of IOAM-Trace-Type. Thus, every transit node that processes this trace option only adds a single data field, which is the Hop_Lim and node_id data field. A transit node that receives a packet with an IOAM trace option that has the Loopback flag set and the IOAM-Trace-Type is not equal to '1' in the most significant bit and '0' in the rest of the bits **MUST NOT** loop back a copy of the packet. The reason for allowing only a single data field per hop is to minimize the impact of amplification attacks.

IOAM encapsulating nodes **MUST NOT** push an IOAM encapsulation with the Loopback flag onto data packets that already include an IOAM encapsulation. This requirement is intended to prevent IOAM Loopback nesting where looped-back packets may be subject to loopback in a nested IOAM-Domain.

4.1.1. Loopback Packet Selection

If an IOAM encapsulating node incorporates the Loopback flag into all the traffic it forwards, it may lead to an excessive amount of looped back packets, which may overload the network and the encapsulating node. Therefore, an IOAM encapsulating node that supports the Loopback flag **MUST** support the ability to incorporate the Loopback flag selectively into a subset of the packets that are forwarded by it.

Various methods of packet selection and sampling have been previously defined, such as [RFC7014] and [RFC5475]. Similar techniques can be applied by an IOAM encapsulating node to apply loopback to a subset of the forwarded traffic.

The subset of traffic that is forwarded or transmitted with a Loopback flag **SHOULD NOT** exceed 1/N of the interface capacity on any of the IOAM encapsulating node's interfaces. This requirement applies to the total traffic that incorporates a Loopback flag, including traffic that is forwarded by the IOAM encapsulating node and probe packets that are generated by the IOAM encapsulating node. In this context, N is a parameter that can be configurable by network operators. If there is an upper bound, M, on the number of IOAM transit nodes in any path in the network, then configuring N such that N >> M (i.e., N is much greater than M) is **RECOMMENDED**. The rationale is that a packet that includes the Loopback flag triggers a looped-back packet from each IOAM transit node along the path for a total of M looped-back packets. Thus, if N >> M, then the number of looped-back packets is significantly lower than the number of data packets forwarded by the IOAM encapsulating node. It is **RECOMMENDED** that the default value of N satisfies N>100 to be used in the absence of explicit operator configuration or if there is no prior knowledge about the network topology or size.

An IOAM-Domain in which the Loopback flag is used **MUST** be configured such that there is expected to be a return path from each of the IOAM transit and IOAM decapsulating nodes; if this expectation does not apply, or if the encapsulating node's identity is not available in the encapsulation header, then configuration **MUST NOT** enable the Loopback flag to be set.

4.2. Receiving and Processing Loopback

A Loopback flag that is set indicates to the transit nodes processing this option that they are to create a copy of the received packet and send the copy back to the source of the packet. In this context, the source is the IOAM encapsulating node and it is assumed that the source address is available in the encapsulation header. Thus, the source address of the original packet is used as the destination address in the copied packet. If IOAM is used over an encapsulation that does not include the address of the encapsulating node, then the transit/decapsulating node does not loop back a copy of the original packet. The address of the node performing the copy operation is used as the source address; the specific method of source address assignment is encapsulation specific, e.g., if an IPv6 encapsulation is used, then the source address can be assigned as specified in [RFC6724]. The copy is also truncated, i.e., any payload that resides after the IOAM option(s) is removed before transmitting the looped-back packet back towards the encapsulating node. Creating the copy that is looped back, and specifically the truncation, may require some encapsulation-specific updates in the encapsulation header. The original packet continues towards its destination. The L-bit **MUST** be cleared in the copy of the packet that a node sends back towards the source.

An IOAM node that supports the reception and processing of the Loopback flag **MUST** support the ability to limit the rate of the looped-back packets. The rate of looped-back packets **SHOULD** be limited so that the number of looped-back packets is significantly lower than the number of packets that are forwarded by the device. The looped-back data rate **SHOULD NOT** exceed 1/N of the interface capacity on any of the IOAM node's interfaces. Using N>100 is **RECOMMENDED**. Depending on the IOAM node's architecture considerations, the loopback response rate may be limited to a lower number in order to avoid overloading the IOAM node.

4.3. Loopback on the Return Path

On its way back towards the source, the copied packet is processed like any other packet with IOAM information, including adding requested data at each transit node (assuming there is sufficient space).

4.4. Terminating a Looped-Back Packet

Once the return packet reaches the IOAM-Domain boundary, IOAM decapsulation occurs as with any other packet containing IOAM information. Note that the looped-back packet does not have the L-bit set. The IOAM encapsulating node that initiated the original loopback packet recognizes a received packet as an IOAM looped-back packet by checking the Node ID in the Hop_Lim/node_id field that corresponds to the first hop. If the Node ID and IOAM-Namespace match the current IOAM node, it indicates that this is a looped-back packet that was initiated by the current IOAM node and processed accordingly. If there is no match in the Node ID, the packet is processed like a conventional IOAM-encapsulated packet.

Note that an IOAM encapsulating node may be either an endpoint (such as an IPv6 host) or a switch/router that pushes a tunnel encapsulation onto data packets. In both cases, the functionality that was described above avoids IOAM data leaks from the IOAM-Domain.

Specifically, if an IOAM looped-back packet reaches an IOAM boundary node that is not the IOAM node that initiated the loopback, the node does not process the packet as a loopback; the IOAM encapsulation is removed, preventing IOAM information from leaking out from the IOAM-Domain. Since the packet does not have any payload, it is terminated.

5. Active Measurement with IOAM

Active measurement methods [RFC7799] make use of synthetically generated packets in order to facilitate measurement. This section presents use cases of active measurement using the IOAM Active flag.

The Active flag indicates that a packet is used for active measurement. An IOAM decapsulating node that receives a packet with the Active flag set in one of its Trace options must terminate the packet. The Active flag is intended to simplify the implementation of decapsulating nodes by indicating that the packet should not be forwarded further. It is not intended as a replacement for existing active OAM protocols, which may run in higher layers and make use of the Active flag.

An example of an IOAM deployment scenario is illustrated in Figure 2. The figure depicts two endpoints: a source and a destination. The data traffic from the source to the destination is forwarded through a set of network devices, including an IOAM encapsulating node (which incorporates one or more IOAM options), a decapsulating node (which removes the IOAM options), and optionally one or more transit nodes. The IOAM options are encapsulated in one of the IOAM encapsulation types, e.g., [IOAM-NSH] or [IOAM-IPV6-OPTIONS].

```
+--------+     +--------+     +--------+     +--------+     +--------+
|        |     |  IOAM  |.....|  IOAM  |.....|  IOAM  |     |        |
+--------+     +--------+     +--------+     +--------+     +--------+
| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |
+--------+     +--------+     +--------+     +--------+     +--------+
  Source     Encapsulating    Transit     Decapsulating   Destination
                  Node          Node           Node

               <------------  IOAM-Domain  ------------>
```

Figure 2: Network Using IOAM

This document focuses on three possible use cases of active measurement using IOAM. These use cases are described using the example of Figure 2.

Endpoint active measurement:
: synthetic probe packets are sent between the source and destination, traversing the IOAM-Domain. Since the probe packets are sent between the endpoints, these packets are treated as data packets by the IOAM-Domain and do not require special treatment at the IOAM layer. Specifically, the Active flag is not used in this case and the IOAM layer does not need to be aware that an active measurement mechanism is used at a higher layer.

IOAM active measurement using probe packets within the IOAM-Domain:
: probe packets are generated and transmitted by the IOAM encapsulating node and are expected to be terminated by the decapsulating node. IOAM data related to probe packets may be exported by one or more nodes along its path by an exporting protocol that is outside the scope of this document (e.g., [IOAM-RAWEXPORT]). Probe packets include a Trace Option that has its Active flag set, indicating that the decapsulating node must terminate them. The specification of these probe packets and the processing of these packets by the encapsulating and decapsulating nodes is outside the scope of this document.

IOAM active measurement using replicated data packets:
: probe packets are created by the encapsulating node by selecting some or all of the en route data packets and replicating them. A selected data packet and its (possibly truncated) copy is forwarded with one or more IOAM options while the original packet is forwarded normally without IOAM options. To the extent possible, the original data packet and its replica are forwarded through the same path. The replica includes a Trace Option that has its Active flag set, indicating that the decapsulating node should terminate it. The current document defines the role of the Active flag in allowing the decapsulating node to terminate the packet, but the replication functionality and the functionality of the decapsulating node in this context is outside the scope of this document.

If the volume of traffic that incorporates the Active flag is large, it may overload the network and the IOAM node(s) that process the active measurement packet. Thus, the rate of the traffic that includes the Active flag **SHOULD NOT** exceed 1/N of the interface capacity on any of the IOAM node's interfaces. Using N>100 is **RECOMMENDED**. Depending on the IOAM node's architecture considerations, the rate of Active-enabled IOAM packets may be limited to a lower number in order to avoid overloading the IOAM node.

6. IANA Considerations

IANA has allocated the following bits in the "IOAM Trace-Flags" registry as follows:

Bit 1	"Loopback" (L-bit)
Bit 2	"Active" (A-bit)

This document is specified as the "Reference" in the registry for both bits.

Note that bit 0 is the most significant bit in the "IOAM Trace-Flags" registry. This bit was allocated by [RFC9197] as the 'Overflow' bit.

7. Performance Considerations

Each of the flags that are defined in this document may have performance implications. When using the loopback mechanism, a copy of the data packet is sent back to the sender (thus, generating more traffic than originally sent by the endpoints). Using active measurement with the Active flag requires the use of synthetic (overhead) traffic.

Each of the mechanisms that use the flags above has a cost in terms of the network bandwidth and may potentially load the node that analyzes the data. Therefore, it **MUST** be possible to use each of the mechanisms on a subset of the data traffic; an encapsulating node needs to be able to set the Loopback and Active flags selectively in a way that considers the effect on the network performance, as further discussed in Sections 4.1.1 and 5.

Transit and decapsulating nodes that support loopback need to be able to limit the looped-back packets (as discussed in Section 4.2) so as to ensure that the mechanisms are used at a rate that does not significantly affect the network bandwidth and does not overload the source node in the case of loopback.

8. Security Considerations

The security considerations of IOAM in general are discussed in [RFC9197]. Specifically, an attacker may try to use the functionality that is defined in this document to attack the network.

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their effect. This is a fundamental assumption with respect to the security aspects of IOAM as further discussed in [RFC9197]. However, even given this limited scope, security threats should still be considered and mitigated. Specifically, an attacker may attempt to overload network devices by injecting synthetic packets that include an IOAM Trace Option with one or more of the flags defined in this document. Similarly, an on-path attacker may maliciously set one or more of the flags of transit packets.

Loopback flag:

an attacker that sets this flag, either in synthetic packets or transit packets, can potentially cause an amplification since each device along the path creates a copy of the data packet and sends it back to the source. The attacker can potentially leverage the Loopback flag for a DDoS attack as multiple devices send looped-back copies of a packet to a single victim.

Active flag:

the impact of synthetic packets with the Active flag is no worse than synthetic data packets in which the Active flag is not set. By setting the Active flag in en route packets, an attacker can prevent these packets from reaching their destination since the packet is terminated by the decapsulating device. However, note that an on-path attacker may achieve the same goal by changing the destination address of a packet. Another potential threat is amplification; if an attacker causes transit switches to replicate more packets than they are intended to replicate (either by setting the Active flag or by sending synthetic packets), then traffic is amplified, causing bandwidth degradation. As mentioned in Section 5, the specification of the replication mechanism is not within the scope of this document. A specification that defines the replication functionality should also address the security aspects of this mechanism.

Some of the security threats that were discussed in this document may be worse in a wide area network in which there are nested IOAM-Domains. For example, if there are two nested IOAM-Domains that use loopback, then a looped-back copy in the outer IOAM-Domain may be forwarded through another (inner) IOAM-Domain and may be subject to loopback in that (inner) IOAM-Domain, causing the amplification to be worse than in the conventional case.

In order to mitigate the performance-related attacks described in Section 7, it should be possible for IOAM-enabled devices to selectively apply the mechanisms that use the flags defined in this document to a subset of the traffic and to limit the performance of synthetically generated packets to a configurable rate. Specifically, IOAM nodes should be able to:

- Limit the rate of IOAM packets with the Loopback flag (IOAM encapsulating nodes) as discussed in Section 4.1.1.
- Limit the rate of looped back packets (IOAM transit and decapsulating nodes) as discussed in Section 4.2.
- Limit the rate of IOAM packets with the Active flag (IOAM encapsulating nodes) as discussed in Section 5.

As defined in Section 4, transit nodes that process a packet with the Loopback flag only add a single data field and truncate any payload that follows the IOAM option(s), thus significantly limiting the possible impact of an amplification attack.

9. References

9.1. 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>.

[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>.

9.2. Informative References

[IOAM-IPV6-OPTIONS]

Bhandari, S., Ed. and F. Brockners, Ed., "In-situ OAM IPv6 Options", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-ipv6-options-09, 11 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-ipv6-options-09>.

[IOAM-NSH] Brockners, F., Ed. and S. Bhandari, Ed., "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", Work in Progress, Internet-Draft, draft-ietf-sfc-ioam-nsh-11, 30 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-sfc-ioam-nsh-11>.

[IOAM-RAWEXPORT] Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", Work in Progress, Internet-Draft, draft-spiegel-ippm-ioam-rawexport-06, 21 February 2022, <https://datatracker.ietf.org/doc/html/draft-spiegel-ippm-ioam-rawexport-06>.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <https://www.rfc-editor.org/info/rfc5475>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <https://www.rfc-editor.org/info/rfc6291>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC7014] D'Antonio, S., Zseby, T., Henke, C., and L. Peluso, "Flow Selection Techniques", RFC 7014, DOI 10.17487/RFC7014, September 2013, <https://www.rfc-editor.org/info/rfc7014>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

Acknowledgments

The authors thank Martin Duke, Tommy Pauly, Donald Eastlake, Paul Kyzivat, Bernard Aboba, Greg Mirsky, and other members of the IPPM working group for many helpful comments.

Contributors

The Editors would like to recognize the contributions of the following individuals to this document.

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
United States of America
Email: sramesh@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

Aviv Kfir
Nvidia
Email: avivk@nvidia.com

Jennifer Lemon
Broadcom
270 Innovation Drive
San Jose, CA 95134
United States of America
Email: jennifer.lemon@broadcom.com

Authors' Addresses

Tal Mizrahi
Huawei
Israel
Email: tal.mizrahi.phd@gmail.com

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

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

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