A Light Weight IOAM for SRv6 Network Programming
draft-li-spring-light-weight-srv6-ioam-01

Abstract

In-situ OAM(IOAM) records OAM information within the packet while the packet traverses a particular network domain. A discussion of the motivation and requirements for in-situ OAM can be found in [I-D.brockners-inband-oam-requirements].

This document defines a light weight IOAM method for SRv6 network programming. In this method, IOAM information is carried in the data packet by the SIDs in the SRH.

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 December 29, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Segment routing (SR) [RFC8402] is a source routing paradigm that explicitly indicates the forwarding path for packets at the source node by inserting an ordered list of instructions, called segments. A segment can represent a topological or service-based instruction.

When segment routing is deployed on IPv6 dataplane, called SRv6 [I-D.ietf-6man-segment-routing-header], a segment is a 128 bit value, and it can be an IPv6 address of a local interface but it does not
have to. As defined in [I-D.ietf-spring-srv6-network-programming], a segment has the format of LOC: FUNCT. The most significant L bits is LOC that is routable and leads packets to the SID originating node. The least significant 128-L bits is the value of FUNCT that defines the local actions associated to the SID. L is the length of LOC and it is flexible. For supporting SR, a new type of routing header, called Segment Routing Header (SRH), which contains a list of SIDs and other needed information, has been defined in [I-D.ietf-6man-segment-routing-header].

In-situ OAM(IOAM) records OAM information within the packet while the packet traverses a particular network domain. A discussion of the motivation and requirements for in-situ OAM can be found in [I-D.brockners-inband-oam-requirements]. [I-D.ali-6man-spring-srv6-oam] defines an IOAM mechanism for SRv6.

However, recording IOAM data in SRH TLVs will bring bigger overhead, which will reduce transport efficiency. In addition, the length of the header will increase in the incremental trace option [I-D.ietf-ippm-ioam-data], which may bring MTU problem and increase the difficulty of packet processing.

This document introduces a light weight IOAM method for SRv6 network programming. In this method, the OAM information is in the segment list of the SRH once the segment is processed.

In most cases, an SRv6 segment will not be reused again after it has been processed for updating the IPv6 destination address (as part of the SRH procedures described in [I-D.ietf-6man-segment-routing-header]). However, these processed SIDs will still be carried in the SRH to the destination of the packet (or penultimate node if PSP is enabled). Therefore, these processed SIDs (i.e.: the 128 bit space they occupy) can be reused for other purposes such as carrying performance measurement information or IOAM [I-D.ietf-ippm-ioam-data] information.

Using the SID in order to carry OAM information allows not to increase the size of the packet header and, of course, will cause the loss of the original SID value.

2. Terminology

This memo makes use of the terms defined in [RFC8402], [I-D.ietf-ippm-ioam-data] and [I-D.ietf-spring-srv6-network-programming].
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.

3. SRv6 Light Weight IOAM

This document defines a light weight IOAM model for SRv6 network programming.

In SRv6, a SID will not be used again after it has been processed according to SRH procedures described in [I-D.ietf-6man-segment-routing-header]. Therefore, the 128 bits of the used SID are reused in order to store metadata such as IOAM data.

In this document, we assume that the rewritable length of a SID is 64 bits at least, since IOAM data need to meet the accuracy requirement as defined at [I-D.ietf-ippm-ioam-data]. The rewritable part consists of the last 64 bits of the SID, which MAY be the FUNCT part.

In order to determine to which node the IOAM data is related, the LOC that identifies the node must remain, especially when the type of IOAM data is not node ID [I-D.ietf-ippm-ioam-data]. For getting rid of the limitation of keeping LOC part of the SID, a Path Segment [I-D.li-spring-srv6-path-segment] is included in the SID list.

In order to indicate the type of IOAM data, the document defines a new field of the SID called the FLAG. With the FLAG field, the format of the SID is LOC:FLAG:FUNCT. The new FLAG field is used in order to indicate additional operations, such as IOAM operations. The IOM data stored in the SID is structured in the following format: <FLAG><IOAMdata>. The procedure will be described in section 5.

3.1. The FLAG Field of SID

In order to indicate the type of IOAM data, this document defines a new field of the SID, called FLAG.

This document does not limit the offset and length of the FLAG field, and it can be configured by the operator. For instance, a FLAG field can be a 8 bits value between the LOC and the FUNCT fields. The offset can be the 48th bit. In other words, 0-47 bit is the LOC, 48-55 bit is the FLAG, and 56-127 bit is the FUNCT. The value of the FLAG field indicates the IOAM processing and IOAM data type, and may have the following values:
0: Non IOAM
1: Timestamp (32-bit seconds and 32-bit subseconds)
2: Packet counter
3: Queue depth
4: Ingress_if_id and egress_if_id (short format)
5: Hop_Lim and node_id
6: Namespace specific data
7: Buffer occupancy
8: Checksum Complement
9-255: Reserved

The IOAM data reuses the format defined in [I-D.ietf-ippm-ioam-data].
A packet counter is a 64-bit value that records the total number of packets in a flow, path or SR policy received by the node. It can be used for use cases like packet loss measurement.

4. Capabilities and SID Format Advertisement

In order to support light weight IOAM, nodes SHOULD advertise the IOAM capability to other nodes within an SR domain via, e.g., IGP extensions. The definition, advertisement and processing of such capability advertisement is out of scope of this document, and will be described in a separate document.

The format of the SID SHOULD also be advertised to the other nodes in the SR domain so that each node having to insert IOAM data, know which format of the SID it has to use (i.e.: the size of the LOC, FLAG and FUNC fields). The description of the advertisement of the SID format is out of scope of this document, and will be described in another document.

5. SIDs Distribution

It has to be noted that the FLAG field does not introduce any new type of SID in the SR architecture.

A node can distribute all variants of SIDs with different FLAG values. For example, Node A instantiates an End SID [I-D.ietf-spring-srv6-network-programming] as
A::0::100(LOC:FLAG:FUNCT), and it supports adding IOAM data of timestamp and queue depth. Then, node A can distribute the SIDs A::0::100, A::1::100, and A::3::100 in order to support respectively non-IOAM End SID, End SID with timestamp recording, and End SID with Queue depth recording. An Ingress node B can use these three variants of SIDs according to the SR policy in order to achieve IOAM with the specific node data.

Alternatively, a node can distribute only a default variant of SID in order to reduce the SID distribution flooding traffic. The other variants can be generated and used by the ingress node according to the capabilities information of the SID endpoint node A. The details will be discussed in another document.

6. Packet Processing

[I-D.ietf-6man-segment-routing-header] describes SRv6 packet processing at the SR source, Transit and SR segment endpoint nodes. This section describes the SRv6 packet processing with the new FLAG field in the SID.

6.1. Source SR Node

This document assumes that, in an operator network, the packet received at the ingress node is encapsulated into an outer IPv6+SRH header. Therefore, the term "ingress" and "source" are to be intended as the same node.

A source node steers a packet into an SR Policy that consists of a segment list. When deploying IOAM, the source node SHOULD insert the associated SID variant into the segment list as illustrated in section 5. The variants of the SID can be learned from SIDs distribution or generated according to node’s capabilities.

After the first SID (SID-List[n-1], last entry in the SID list) is updated to the IPv6 DA, the source node SHOULD rewrite the SID with IOAM data according to the value of FLAG field and then send it to the next hop.

6.1.1. Reduced SRH

Reduced SRH cannot support light weight IOAM since there is no space for carrying IOAM data of the source node.
6.2. Transit Node

As specified in [RFC8200], the only node allowed to inspect the Routing Extension Header (and therefore the SRH), is the node corresponding to the DA of the packet. Any other transit node MUST NOT inspect the underneath routing header and MUST forward the packet toward the DA according to its IPv6 routing table. Thus, there is no modification of packet processing on transit nodes.

6.3. SR Segment Endpoint Node

As per [I-D.ietf-6man-segment-routing-header], when an SRv6-capable node receives an IPv6 packet, it performs a longest-prefix-match lookup on the packets destination address. When this lookup returns a FIB entry that represents a locally instantiated SRv6 SID, then the node should process the SRH and related SIDs.

As per [I-D.ietf-spring-srv6-network-programming], if the IPv6 DA is a local SID instantiated by the node, then it should be looked up in "My local SID table" in order to execute the instruction bound to it. The "My Local SID Table" matches on a per longest-prefix-match basis.

In order to process FLAG field, there are two options:

- All variants of the SID should be generated by the endpoint node and stored in the "My Local SID Table" so that the variants of the SID in IPv6 DA can match the entry by longest-prefix-match basis. The node looks up the variant SID and executes the associated instruction. IOAM data will be rewritten into the SID after the SID has been copied into the DA of the packet. In this way, all variants will be instantiated in the SID table.

- The endpoint node only distributes a default variant SID (FLAG value is 0) and stores it in the "My Local SID Table". Before looking up the SID, the value of the FLAG part is obtained and then make it as all zero. The node looks up the default variant SID to get the instruction. The FLAG related instruction that sets IOAM data into the SID is executed after the DA of the packet is updated with the SID (as per SRH processing - [I-D.ietf-spring-srv6-network-programming]).

6.3.1. Egress Node

In this document it is assumed that the egress node is the final destination of the encapsulated packet. Therefore, the egress node removes the outer IPv6 and SRH header.
As the last SRV6 endpoint node in the SID list, the egress node cannot write any IOAM data into the SID since the segment list is completed and there is no more SID to use. Therefore, the egress node should record the related IOAM data and punt the data with a copied packet to the CPU for further IOAM processing.

6.4. Instructions for FLAG

In order to achieve lightweight IOAM, processing of the FLAG field is required and illustrated by the pseudo code here below:

1. If FLAG == n:
2.   Get the IOAM-data D1 of type n.
3.   If SL=1 and DA is End.S or if SL=0:  
4.       Punt a copied packet with D1 to CPU for further processing; ;Ref2
5.   Else:
6.     Insert following code after the instruction of updating DA:
7.       Update SRH[SL][64:] with D1  ; ;Ref3

- Ref1: If SL is 0 or SL is 1 and the DA is an End.S [I-D.ietf-spring-srv6-network-programming], it means that the current node is the egress node. The node will punt a copied packet with the D1 to the CPU for further processing.
- Ref2: At the egress node, in order not to affect the packet forwarding, a copy of the data packet should be punted instead of the data packet itself. However, this document does not limit implementations to punt the entire packet or only headers of the packet.
- Ref3: If the node is not the egress node the node only inserts "updates the SID with D1" after the instruction of "update DA with SRH[SL]". The last 64 bits of SID SRH[SL] is rewritten with D1 after updating SID SRH[SL] to IPv6 DA. If there is no Path Segment in the segment list, the locator part needs to remain for identifying a SID. The related IOAM data at the ingress node should be written into the SID before sending the data packet.

The pseudo code below illustrates the example where the FLAG is set to 1, indicating that the IOAM data is a timestamp:
1. If FLAG == 1:
2.  Get the receiving timestamp T1.
3.  If SL=1 and DA is End.S or if SL=0:
4.    Punt a copy of the packet with T1 to CPU for further processing
5.  Else:
6.     Insert following code after the instruction of updating DA:
7.         Update SRH[SL][64:] with T1

7. Illustrations

For easy understanding, the following simple topology is used for illustration.

CE-A ---- N1-----N2-----N3---- CE-B

Reference Topology

In the reference topology:
o Nodes N1, N2, and N3 are all SRv6 capable nodes.
o Node N1 and N3 are configured with a tenant 10, each connected to
CE-A and CE-B respectively.
o Node Nk has Ak::/48 for its local SID space from which Local SIDs
are explicitly allocated.
o A2::1::1 is an End [I-D.ietf-spring-srv6-network-programming]
function with timestamp recording allocated by N2
o A3::1::D10 is an End.DT4
[I-D.ietf-spring-srv6-network-programming] function with timestamp
recording bound to tenant IPv4 table 10 allocated by N3.
o Bk::/48 is the loopback address of node K for IGP.

It is assumed that the measured flow from CE-A to CE-B will travel
along the path <N1, N2, N3>.

When the ingress node N1 receives an IPv4 packet with SA:CE-A, and
DA:CE-B, the ingress node N1 will encapsulate the packet into an IPv6
header followed by an SRH with the SID list <A2::1::1, A3::1::D10>,
so that the packet header is now (B1, A2::1::1) (A3::1::D10,
A2::1::1, SL=2)(CE-A, CE-B). The DA of the IPv6 header is then
updated with the first segment.
7.1. Timestamp Recording Procedures

After updating the DA with A2::1::1, the ingress node N1 updates the SID A2::1::1 in the segment list with current timestamp. Then it forwards the packet to N2 according to its RIB/FIB. Assuming that the timestamp value is T1, then the updated packet header becomes (B1, A2::1::1) (A3::D10, A2::1::T1, SL=1) (CE-A, CE-B).

When N2 receives the packet with DA A2::1::1 which is an End SID with timestamp recording originated by N2, N2 will insert "Update SRH[SL][64:] with the current timestamp" after instruction of "update DA with SRH[SL]", and then process this End SID.

After updating IPv6 DA with SRH[0] A3::1::D10, the current timestamp will be recorded at the last 64 bits of End SID A3::1::D10. Assuming that the timestamp value is T2, the updated packet header becomes (B1, A3::1::D10) (A3::1::T2, A2::1::T1, SL=0) (CE-A, CE-B). According to the updated DA A3::1::D10, the packet will be forwarded to N3.

When the egress node N3 receives the packet with header (B1, A3::1::D10) (A3::1::T2, A2::1::T1, SL=0) (CE-A, CE-B), N3 should timestamp the copied packet and punt it to CPU for further processing. Then, N3 decapsulates the outer header and forwards the inner packet to CE-B based on the routes in tenant-10 IPv4 table.

IOAM data processing can be implemented by a node or a remote controller. In this solution, only one receiving timestamp or sending timestamp can be recorded in the SID at each endpoint node. Therefore, it is RECOMMENDED that the ingress node records the sending timestamp while the other nodes record the receiving timestamp.

8. Backward Compatibility Considerations

TBA

9. IANA Considerations

TBA

10. Security Considerations

TBA
11. Contributors

TBA

12. Acknowledgements

Many thanks to Stefano’s professional comments.

13. References

13.1. Normative References

[I-D.ietf-6man-segment-routing-header]

[I-D.ietf-spring-srv6-network-programming]

[I-D.li-spring-srv6-path-segment]


13.2. Informative References


[I-D.brockners-inband-oam-requirements] Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., Lapukhov, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.


Authors’ Addresses

Cheng Li
Huawei

Email: chengli13@huawei.com

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com