

loops
Internet-Draft
Intended status: Informational
Expires: September 10, 2020

J. Wang
S. Nie
B. Lei
China Telecom
C. Bormann
Universitaet Bremen TZI
March 09, 2020

Embedding LOOPS in SRv6
draft-wang-loops-srv6-binding-00

Abstract

LOOPS (Local Optimizations on Path Segments) aims to provide local in-network loss recovery. It can be used with tunneling protocols to efficiently recover lost packets on a single segment of an end-to-end path instead of leaving recovery to the end-to-end protocol, traversing the entire path.

Segment Routing (SR) leverages the source routing mechanisms and steers the packets through a policy instantiated as an ordered list of instructions called segments. LOOPS can be embedded in an SR segment to improve the packet recovery. [draft-welzl-loops-gen-info](#) defines the generic information model, without binding that to any specific protocols, to be carried between LOOPS ingress and egress nodes, which can be SR segment endpoints. This document specifies the concrete mechanisms to embed LOOPS in SRv6 segment.

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 September 10, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	LOOPS TLV in SRv6	4
3.1.	Flags and Flag Based Data	5
4.	Embedding LOOPS in SRv6	6
5.	Security Considerations	8
6.	IANA Considerations	8
7.	References	8
7.1.	Normative References	8
7.2.	Informative References	9
	Acknowledgements	9
	Authors' Addresses	10

[1.](#) Introduction

LOOPS (Local Optimizations on Path Segments) aims to provide local in-network loss recovery. [[I-D.li-tsvwg-loops-problem-opportunities](#)] shows an overview on the problems that LOOPS could address and some use cases where LOOPS can be employed to improve the performance. When an end-to-end path contains one or more LOOPS segments, packet loss recovery can be performed over such a segment. Such in-network recovery is faster than the end-to-end packet recovery in most cases, especially when the LOOPS enabled segment is a small part of the entire path in terms of latency, or when it occasionally suffers microburst losses. In addition, with the increasing deployment of encrypted transport layer like QUIC, the traditional performance enhancing proxies (PEPs, [[RFC3135](#)]) are no longer applicable. LOOPS enabled on a network segment can work with no dependency on transport layer or higher information. Thus it can improve performance over a path with segments of varying quality.

[[I-D.welzl-loops-gen-info](#)] illustrates the generic information to be carried over a LOOPS segment. Such information is used for packet loss detection, packet retransmission (if required), segment congestion indication to end-to-end congestion control loop, etc. In typical cases where the segment is tunnel based, this information is embedded in the tunnel encapsulation. [[I-D.welzl-loops-gen-info](#)] does not exhaustively describe how each protocol specifically carries LOOPS information and what are the encapsulations.

[[I-D.bormann-loops-geneve-binding](#)] is an example for a binding of LOOPS to a specific encapsulation, in this case Geneve. Similarly, the present document is a binding of LOOPS to SRv6.

Segment Routing (SR) [[RFC8402](#)] uses source routing techniques to deliver the data through a pre-defined path and/or functions. It can be used to select a traffic path, for instance a lower latency one. SRv6 (SR over IPv6) runs on a best effort IPv6 network, and its segments suffer from different levels of packet loss. An SR segment can naturally be a LOOPS segment to perform in-network packet loss recovery.

There are some typical scenarios that we could use LOOPS to enhance SRv6 data transmission. For example, most mobile payment services have critical requirements of the latency and packet loss over the Internet. The pre-defined SRv6 path could be the shortest one on average, with some occasional packet loss. Such occasional packet loss does not always trigger the selection of a better path as change of a path normally can not be executed in real time. Hence LOOPS can be used to recover the packet loss over SRv6 segments to guarantee such critical services. The ingress endpoint identifies the key services which require LOOPS enablement and the traffic flows through the pre-defined SIDs (segment IDs). There are different ways to identify such services, for instance, using the destination IP address and port. In SR context, it could also be coupled to some special SID when an SRv6 ingress node applies its path selection policy. In-network recovery then can be performed between the endpoints of a SR segment, to be more specific between two SIDs.

This document describes how to embed LOOPS in SRv6 data packets.

2. Terminology

This document makes use of the terminology defined in [[I-D.welzl-loops-gen-info](#)].

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. LOOPS TLV in SRv6

SRv6 [[I-D.ietf-6man-segment-routing-header](#)] defines meta-data in TLV format for segment processing. SRv6 segment endpoints are LOOPS ingress and egress. In the rest of this document, the term "segment" is used interchangeably both for the LOOPS segment and the SRv6 segment.

A new TLV called LOOPS is defined in this document. Figure 1 shows its format.

Alignment requirement: 4n

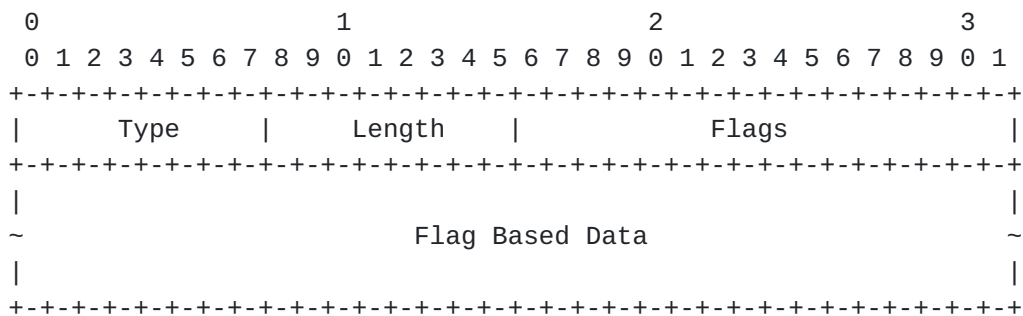


Figure 1: SRv6 LOOPS TLV

where:

- o Type: to be assigned by IANA (suggested value 128). The value should be at least 128 as LOOPS TLV data may change en route to the packet's final destination. Its highest-order bit of Type must be 1.
- o Length: The length of the variable length data in bytes. The maximum total length of this TLV is $(8 + 127) = 135$ bytes.
- o Flags: 16 bits, as described in [Section 3.1](#). Some of the flags indicate the presence of additional data in the field of Flag Based Data.
- o Flag Based Data: This field consists of one or multiple optional data blocks whose presence is indicated by the corresponding flag bits. Please see [Section 3.1](#) for details.

3.1. Flags and Flag Based Data

Flags are defined in Figure 2. Some flags have additional data blocks in Flags Based Data field. Those additional data blocks are placed in order of flags. Figure 2 shows the format and definition of flags defined.

```

+---+---+---+---+---+---+---+---+---+---+
|M|I|R|D|S|T|E|A|R|           |B|
+---+---+---+---+---+---+---+---+---+---+

```

Figure 2: Flags in SRv6 LOOPS TLV

- o M: Mode flag.
 - * 0: Retransmission mode. In this mode, the LOOPS TLV format and operations follow this document.
 - * 1: FEC mode.
- o I: Initial Packet Sequence Number (PSN) flag. It is set by the LOOPS ingress to notify the egress about using a new initial PSN.
- o R: Initial PSN Received flag; echo of I flag provided by the LOOPS egress.
- o D: ACK Desired flag. It is set by the LOOPS ingress if it wants the egress to generate an acknowledgement immediately upon receiving a particular packet.
- o S: PSN flag. When set, it indicates a PSN data block is carried in Flag Based Data field. It must be set when a packet payload is present. It must not be set if the packet is a pure LOOPS ACK packet, i.e. when no payload is included in the packet.
- o T: Timestamp flag. When set, it indicates a Timestamp data block is carried in the Flag Based Data field.
- o E: Echoed Timestamp flag. When set, it indicates an Echoed Timestamp data block is carried in the Flag Based Data field.
- o A: ACK number flag. When set, it indicates the presence of a Block 1 ACK information block.
- o R: Reception time flag: May only be set if A is set. Indicates that an absolute reception time is given (Format TBD).

Finally, a single flag bit is defined that causes the addition of a variable-length block (therefore this flag is put as the least significant bit of Flags):

- o B: Block 2 flag. When set, it indicates the presence of a Block 2 ACK information block, with the following format: TBD

((We borrowed most text about encapsulation from [draft-bormann-loops-geneve-binding](#). 1. M flag for Mode is put in flags field which is different from Geneve encap. 2. Further discussion about a minimum and unified set of loops format is required since it turns out most encap format are the same for Geneve and SRv6.)))

4. Embedding LOOPS in SRv6

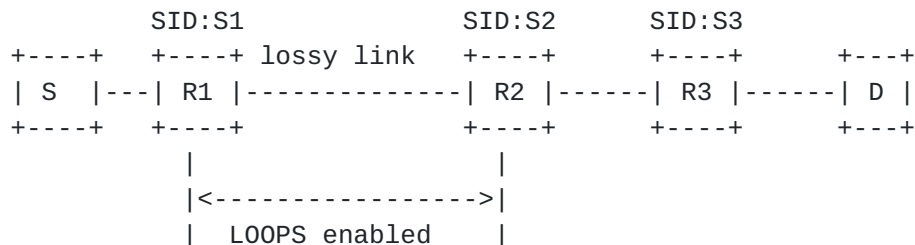
LOOPS can be enabled on any segment in SRv6 deployment. For instance, a SID list <S1, S2, S3> may enable LOOPS on any segments independently. If segment S2 is subject to higher packet loss, LOOPS is enabled on S2 without impact on S1 and S3.

Figure 3 shows packets sent from S to D in an SRv6 domain via SID list <S1, S2, S3> and LOOPS is enabled between R1 and R2. The configuration needs to make sure both R1 and R2 can support LOOPS TLV.

Format

(SA,DA): (src address, dst address)

(S3, S2, S1; SL):(SID list; Segments Left)



(S, S1)

(D, S3, S2, S1; 3)

----->

(S, S2)

(D, S3, S2, S1; 2)

+ L00PS TLV

-----> (S, S3)

(D, S3, S2, S1; 1)

(S2, S1)

----->

(S1; 0)

(S, D)

+ L00PS TLV

(D, S3, S2, S1; 0)

<-----

----->

Figure 3: Segment enabled L00PS in SRv6

L00PS TLV should be parsed and removed by the destined SRv6 SID. That is to say L00PS is enabled on a segment independently.

In the forward path, a L00PS TLV is added by SR segment endpoint R1 and sent to SID S2. When R2 receives the packet with SRH and L00PS TLV present, it should get the previous segment SID which is S1 as shown in Figure 3 from field Segment List[Segment Left + 1] where Segment Left is field value in the received SRH. When the value of Segment Left is equal to the value of Last Entry in SRH, the previous segment SID is set to the value of the source IPv6 address in IP header.

Reduced SRH (section 4.1.1 in [[I-D.ietf-6man-segment-routing-header](#)]) does not contain the first segment of the related SR policy in Segment List, the first segment is the one already in the DA of the IPv6 header. Hence when reduced SRH is used, if the value of Segment Left is equal to the value of (Last Entry + 1) in SRH, the previous

segment SID is set to the value of the source IPv6 address. A special case that needs to be taken care of is when LOOPS is enabled on the second segment with reduced SRH. As the SID of the first segment is not present in the field of Segment List, there is no direct way to set the right value of previous segment SID (which is SID of the first segment) when the second segment receives the LOOPS enabled packet. Hence it is suggested not to use reduced SRH when LOOPS is in use or the configurations need to ensure that LOOPS is not enabled on the second segment when reduced SRH is in use.

Any generated feedback such as ACKs should be sent as reverse channel information back to previous segment SID. The LOOPS reverse channel information needs to be sent from SID S2 to SID S1 in figure 3. Such information can be piggybacked in data packets in the reverse direction. When piggyback is not possible, a pure LOOPS ACK needs to be sent back. In this case, the reverse information packet has SRH with LOOPS TLV but has no real payload. The field of Next Header in SRH for pure LOOPS ACK should be set to 59 (No Next Header) [[RFC8200](#)].

When the sender and receiver are outside the SRv6 domain, the SR ingress (which could be R1 in Figure 3) will encapsulate the packet received in an outer header with an SRH. LOOPS can be enabled on the segment indicated by the outer header.

5. Security Considerations

The security considerations of [[I-D.welzl-loops-gen-info](#)] and [[I-D.ietf-6man-segment-routing-header](#)] apply.

6. IANA Considerations

IANA is requested to assign a code point value for LOOPS TLV from Segment Routing Header TLVs Registry.

Assigned Value	Description
-----	-----
128	LOOPS (Local Optimizations on Path Segments)

7. References

7.1. Normative References

- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", [RFC 3135](#), DOI 10.17487/RFC3135, June 2001, <<https://www.rfc-editor.org/info/rfc3135>>.

- [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>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [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>>.

7.2. Informative References

- [I-D.li-tsvwg-loops-problem-opportunities]
Yizhou, L., Zhou, X., Boucadair, M., and J. Wang, "LOOPS (Localized Optimizations on Path Segments) Problem Statement and Opportunities for Network-Assisted Performance Enhancement", [draft-li-tsvwg-loops-problem-opportunities-04](#) (work in progress), January 2020.
- [I-D.welzl-loops-gen-info]
Welzl, M. and C. Bormann, "LOOPS Generic Information Set", [draft-welzl-loops-gen-info-02](#) (work in progress), November 2019.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-26](#) (work in progress), October 2019.
- [I-D.bormann-loops-geneve-binding]
Bormann, C., "Embedding LOOPS in Geneve", [draft-bormann-loops-geneve-binding-00](#) (work in progress), November 2019.

Acknowledgements

The authors would like to thank Yizhou Li for her help on LOOPS generic information.

Authors' Addresses

Jianglong Wang
China Telecom

Email: wangjl50@chinatelecom.cn

Shizhong Nie
China Telecom

Email: nieshzh@chinatelecom.cn

Bo Lei
China Telecom

Email: leibo@chinatelecom.cn

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

