

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: November 15, 2026

W. Cheng
L. Gong
China Mobile
C. Lin
New H3C Technologies
Louis Chan
Individual
May 14, 2026

IGP Extensions for Optimized SRv6 SID Advertisement
draft-cheng-lsr-extension-opt-srv6-sid-adv-03

Abstract

When the IGP runs SRv6 Flex-Algo or performs QoS resource allocation, it needs to assign a large number of END.X SIDs, which can significantly impact IGP LSDB advertisements and overall performance.

This document proposes a simplified method for advertising a large number of SRv6 SIDs. This method is particularly useful in scenarios that require generating many END.X SIDs, such as when supporting numerous Flex-Algo algorithms. It helps reduce the size of LSDB advertisements and improves IGP advertisement efficiency and operational performance.

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 November 15, 2026.

Copyright Notice

Copyright (c) 2026 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 (<http://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.....	3
1.1. Conventions and Terminology.....	4
2. Motivation.....	4
3. Solution.....	4
3.1. IS-IS extension.....	6
3.2. OSPFv3 extension.....	7
4. Security Considerations.....	8
5. Compatibility considerations.....	8
6. IANA Considerations.....	9
6.1. IS-IS SRv6 Locator LSA Sub-TLVs.....	9
6.2. OSPFv3 SRv6 Locator LSA Sub-TLVs.....	9
7. References.....	9
7.1. Normative References.....	9
7.2. Informative References.....	10
Authors' Addresses.....	11

1. Introduction

The Segment Routing (SR) allows for a flexible definition of end-to-end paths by encoding paths as sequences of topological sub-paths, called "segments". As defined in [RFC8402] and [RFC8986], an SRv6 Segment Identifier (SID) is an IPv6 address explicitly associated with the segment and consists of Locator, Function and Argument parts.

[RFC9352] defines the SRv6 End SID sub-TLV, the SRv6 End.X SID sub-TLV, and the SRv6 LAN End.X SID sub-TLV in IS-IS.

The SRv6 End SID sub-TLV is used to advertise an SRv6 SID with Endpoint behaviors which do not require a particular neighbor. The SRv6 End.X SID sub-TLV is used to advertise an SRv6 SID associated with a point to point adjacency. The SRv6 LAN End.X SID sub-TLV sub-TLV is used to advertise an SRv6 SID associated with a LAN adjacency. Each of these sub-TLVs contains a complete 128-bit SID and the sub-TLV length is quite long.

Multiple SRv6 End.X SIDs can be associated with the same point to point adjacency or the same physical LAN neighbor. Each SID is advertised in a single SRv6 End.X SID sub-TLV or SRv6 LAN End.X SID sub-TLV. These SIDs are possibly associated to the same Locator, therefore the main differences among the sub-TLVs may be a few bits in the Function part of SID and the Endpoint Behavior value indicating different flavors.

The number of End.X SIDs has a positive correlation with the number of neighbors. Assume that, each neighbor is assigned with End.X SIDs, and each End.X behavior has several different flavors, such as PSP, USP, USD, no PSP/USP/USD, etc. Then, the number of End.X SIDs will be at least the number of neighbors multiplied by the number of flavors.

If Flexible-Algorithm is applied on SRv6 forwarding plane as defined in [RFC9350], a node generally advertises a Flex-Algorithm specific locator for each Flex-Algorithm it participates in and also advertises associated SRv6 END.X SIDs for every link that has not been pruned from the Flex-Algorithm computation.

This document proposes a minimal advertisement method for advertising the bulk generation of specific END.X SIDs. Other routers can use the baseline END.X SID to generate specific END.X SIDs in bulk for particular scenarios and use them in the computation of paths for SR-TE or TI-LFA.

The most common practice is to generate bulk END.X SIDs based on the Flex-Algo algorithm. After generating the END.X SID in algorithm 0, the source device uses it as the baseline END.X SID, directly inherits the Func field from the baseline SID, and generates END.X SIDs in other algorithms. When notifying the END.X SID, the source router only needs to advertise the END.X SID in algorithm 0. Other routers calculate END.X SIDs in other algorithms based on the same algorithm. This method can also be used in other similar scenarios for bulk generating END.X SIDs, such as in HQoS queues scenarios.

1.1. Conventions and Terminology

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. Motivation

In SRv6, the IGP protocol assigns an END.X SID to each neighbor. SRv6 TE or TI-LFA functionality is achieved by specifying the actual path of traffic using END.X SIDs as the path for a specified segment routing.

The IGP protocol advertises neighbor information, which includes the assigned END.X SID for each neighbor. Other routers extract the END.X SID from the neighbor information to obtain the END.X SID associated with that neighbor.

To support Flex-Algo, an END.X SID will be assigned for each Flex-Algo algorithm associated with a neighbor. In scenarios with a large number of neighbors and support for numerous Flex-Algo algorithms, advertising the corresponding END.X SIDs for each neighbor can result in a large LSDB, which in turn can lead to extensive LSDB flooding.

To address this issue, this document describes a mechanism where, for a given neighbor, it is sufficient to advertise the END.X SIDs associated with the common Flex-Algo algorithm and the relationships between the END.X SIDs in other Flex-Algo algorithms and the END.X SIDs in the common Flex-Algo algorithm. This significantly reduces the size of the IGP LSDB and improves operational performance.

3. Solution

Segment Identifier (SID) - A 128-bit IPv6 address that represents the SRv6 instruction. It consists of locator, function, args, and

MBZ parts, where the locator is used to identify the destination node, and the function identifier indicates the specific operation to be performed on packets at that node. As shown in figure 1:

```

+-----+-----+-----+-----+-----+-----+
|  Block   | Node ID |   Function   | Agruments |  MBZ   |
+-----+-----+-----+-----+-----+
|<-----Locator----->|

```

Figure 1: SRv6 SID Structure

In general, each Flex-Algo algorithm specifies an SRv6 Locator, and designates a prefix length via the Locator. When assigning adjacency labels to neighbors for this Flex-Algo algorithm, the specified Locator prefix is followed, and a Function is assigned to form an adjacency label, with the args and MBZ fields typically left unspecified. This approach allows for the allocation of an SRv6 adjacency label for each supported Flex-Algo.

For example, the common algorithm FA0 assigns END.X SID as Locator0.Func0.Args, algorithm FA1 assigns Locator1.Func1.Args, and algorithm FA2 assigns Locator2.Func2.Args. When announcing to the outside, it is necessary to publish the adjacency labels from each Flex-Algo algorithm. The range of Flex-Algo algorithm is from 128 to 255. In the given examples, we have omitted the MBZ section. Please refer to Figure 2 for the specific allocation of adjacency SIDs.

```

+-----+-----+-----+-----+
|Flex-Algo | Locator          |   END.X SID   |
+-----+-----+-----+-----+
|  FA0     | Locator0         | Locator0.Func0.Args |
+-----+-----+-----+-----+
|  FA1     | Locator1         | Locator1.Func1.Args |
+-----+-----+-----+-----+
|  FAx     | Locatorx         | Locatorx.Func2.Args |
+-----+-----+-----+-----+

```

Figure 2: END.X SID Allocation in Flex-Algo

When there are numerous neighbors and support for a large number of Flex-Algo algorithms, announcing a large number of SRv6 adjacency SIDs can significantly impact IGP LSDB advertisements and overall performance.

To address this issue, we propose utilizing the adjacency SID from the common algorithm to automatically calculate the corresponding adjacency SIDs in the respective of the Flex-Algo algorithms.

When announcing the adjacency SIDs externally, we only need to announce the adjacency SIDs corresponding to the common Flex-Algo algorithm. The adjacency SIDs for other Flex-Algo algorithms will be generated based on this information. The method of generation involves inheriting the Func section of the Func section.

Flex-Algo	Locator	END.X SID
FA0 (Common)	Locator0	Locator0.Func.Args
FA1	Locator1	Locator1.Func.Args
FAX	Locatorx	Locatorx.Func.Args

Figure 3: Bulk END.X SID Allocation in Flex-Algo

Furthermore, when advertising the Locator information used by each Flex-Algo, it is necessary to include positions for inheriting the Func segment from the common adjacency SID.

Typically, the length of the Func of the benchmark END.X SID should be the same as the length of the Func of the automatically generated END.X SID.

The specific generation process is as follows: based on the locator address information originally advertised by the algorithm, and using the 16-byte END.X SID mask to perform a bitwise AND operation with the base END.X SID, extract the Func field. Then, combine this Func field with the locator address information to generate a new END.X SID as defined in the algorithm.

Taking IS-IS as an example, assuming there are 100 neighbors, and each interface supports 128 Flex-Algo algorithms. Without batch generation, it would be necessary to advertise 12,800 adjacency SIDs. Assuming each SID occupies 30 bytes and each LSP 1,500 bytes, 256 LSP fragments would need to be generated at this point.

With the introduction of the flex-algo extension, it is only necessary to advertise 128 SID information, resulting in the creation of only 3 LSP fragments.

3.1. IS-IS extension

[RFC9352] defines the format of the Locator TLV. In order to support bulk adjacency SID generation, it is necessary to extend the Locator TLV to include the 16 bytes SID Mask information. Use the 16-byte

END.X SID mask to perform a bitwise AND operation with the base END.X SID and extract the Func field. Then, combine the Func field with the locator address information to generate a new END.X SID.

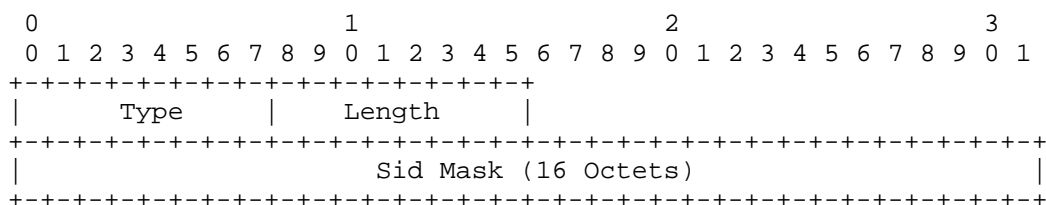


Figure 4: ISIS Adjacent-Sid-Offset Sub-Sub-TLV

where:

- o Type: TBD. Single octet, as defined in Section 9 of [ISO10589].
- o Length: Single octet, as defined in Section 9 of [ISO10589].
- o Sid Mask: 16 octets. Use the 16-byte END.X SID mask to perform a bitwise AND operation with the base END.X SID and extract the Func field. Then, combine the Func field with the locator address information to generate a new END.X SID.

3.2. OSPFv3 extension

[RFC9513] defines the format of the Locator TLV. In order to support bulk adjacency SID generation, it is necessary to extend the Locator TLV to include the 16 bytes SID Mask information. Use the 16-byte END.X SID mask to perform a bitwise AND operation with the base END.X SID and extract the Func field. Then, combine the Func field with the locator address information to generate a new END.X SID.

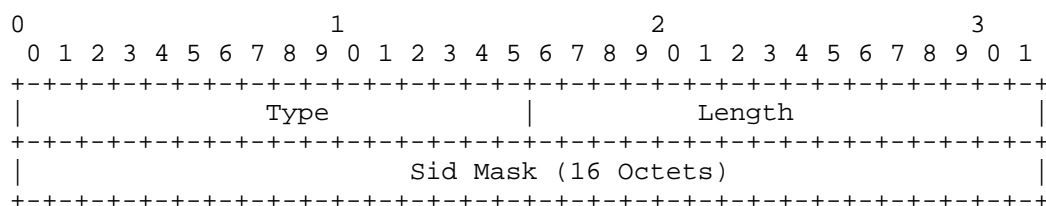


Figure 5: OSPFv3 Adjacent-Sid-Offset Sub-Sub-TLV

where:

- o Type: TBD. 2-octet field.
- o Length: 2-octet field.
- o Sid Mask: 16 octets. Use the 16-byte END.X SID mask to perform a bitwise AND operation with the base END.X SID and extract the Func field. Then, combine the Func field with the locator address information to generate a new END.X SID.

4. Security Considerations

TBD.

5. Compatibility considerations

This functionality consists of two parts:

Firstly, during the allocation of SRv6 END.X SIDs on routers, END.X SIDs are allocated in accordance with the batch allocation rules described in this document.

Secondly, when a router allocates its assigned END.X SIDs for external advertisement, only the END.X SIDs allocated within the base algorithm are explicitly advertised. END.X SIDs allocated within other algorithms are not extensively advertised. Instead, the receiving router generates the END.X SIDs within other algorithms based on the same rules as the assigning router.

The first part of the functionality does not pose any compatibility issues. However, the second part requires support for this functionality on the receiving router. In real deployments, if all routers within the IGP domain that support SRv6 functionality also support this feature, then batch END.X SID advertisement can be performed. Conversely, if some routers do not support this feature, the assigning router needs to advertise all END.X SIDs externally. Therefore, for the sake of compatibility, routers that support this feature should provide a configuration command to disable the bulk advertisement of END.X SIDs and instead advertise detailed information.

6. IANA Considerations

6.1. IS-IS SRv6 Locator LSA Sub-TLVs

This document defines a new sub-sub-tlv for the IS-IS SRv6 Locator TLV.

Type	Description	Reference
TBD	Adjacent-Sid-Offset	This Document

6.2. OSPFv3 SRv6 Locator LSA Sub-TLVs

This document defines a new sub-sub-tlv for the OSPFv3 SRv6 Locator TLV.

Value	Description	Reference
TBD	Adjacent-Sid-Offset	This Document

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017
- [RFC9352] Psenak, P., Ed., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane", RFC 9352, DOI 10.17487/RFC9352, February 2023, <<https://www.rfc-editor.org/info/rfc9352>>.
- [ISO10589] International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", Nov 2002.

7.2. Informative References

- [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>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [RFC9350] Psenak, P., Ed., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", RFC 9350, DOI 10.17487/RFC9350, February 2023, <<https://www.rfc-editor.org/info/rfc9350>>.

Authors' Addresses

Weiqiang Cheng
China Mobile
China

Email: chengweiqiang@chinamobile.com

Liyan Gong
China Mobile
China

Email: gongliyan@chinamobile.com

Changwang Lin
New H3C Technologies
China

Email: linchangwang.04414@h3c.com

Louis Chan
Individual

Email: lwtchan@gmail.com

