Internet-Draft
Intended status: Standards Track
Expires: November 18, 2021

Shaofu. Peng
Ran. Chen
ZTE Corporation
Ketan. Talaulikar
Peter. Psenak
Cisco Systems

May 17, 2021

Algorithm Related IGP-Adjacency SID Advertisement
draft-peng-lsr-algorithm-related-adjacency-sid-03

Abstract

Segment Routing architecture supports the use of multiple routing algorithms, i.e, different constraint-based shortest-path calculations can be supported. There are two standard algorithms: SPF and Strict-SPF, defined in Segment Routing architecture. There are also other user defined algorithms according to Flex-algo applicaiton. However, an algorithm identifier is often included as part of a Prefix-SID advertisement, that maybe not satisfy some scenarios where multiple algorithm share the same link resource. This document complement that the algorithm identifier can be also included as part of a Adjacency-SID advertisement

Table of Contents

1. Introduction

Segment Routing architecture [RFC8402] supports the use of multiple routing algorithms, i.e, different constraint-based shortest-path calculations can be supported. There are two standard algorithms, i.e, SPF and Strict-SPF, that defined in Segment Routing architecture. For SPF, the packet is forwarded along the well known ECMP-aware Shortest Path First (SPF) algorithm employed by the IGPs. However, it is explicitly allowed for a midpoint to implement another forwarding based on local policy. For Strict Shortest Path First (Strict-SPF), it mandates that the packet be forwarded according to

the ECMP-aware SPF algorithm and instructs any router in the path to ignore any possible local policy overriding the SPF decision.

There are also other user defined algorithms according to IGP Flex Algorithm [I-D.ietf-lsr-flex-algo]. IGP Flex Algorithm proposes a solution that allows IGPs themselves to compute constraint based paths over the network, and it also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths. It specifies a set of extensions to ISIS, OSPFv2 and OSPFv3 that enable a router to send TLVs that identify (a) calculation-type, (b) specify a metric-type, and (c)describe a set of constraints on the topology, that are to be used to compute the best paths along the constrained topology. A given combination of calculation-type, metric-type, and constraints is known as an FAD (Flexible Algorithm Definition).

However, an algorithm identifier is often included as part of a Prefix-SID advertisement, that maybe not satisfy some scenarios where multiple algorithm share the same link resource. This document complement that the algorithm identifier can be also included as part of an Adjacency-SID advertisement for SR-MPLS.

2. 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. Use-cases

There are several use-cases for the algorithm-aware Adjacency-SID:

case-1: an SR-TE policy may be instantiated within specific Flex-algo plane, i.e., the SID list may contain algorithm related SIDs. An algorithm-aware Adjacency-SID included in the SID list can not only steer the traffic towards the link, but also apply specific QoS policy for that algorithm.

case-2: a TI-LFA backup path computed in Flex-algo plane may contain Adjacency Segments and require to contain an algorithm-aware Adjacency-SID. An algorithm-aware Adjacency-SID included in the TI-LFA SID list can not only steer the traffic towards the link, but also distinguish traffic between different algorithms.

case-3: for the protected Adjacency-SID which belongs to SR-TE path within specific Flex-algo plane, the backup path of such

Adjacency-SID need follow the algorithm specific constraints that is consistent with the primary SR-TE path.

4. Adjacency Segment Identifier per Algorithm

4.1. ISIS Adjacency Segment Identifier per Algorithm

[RFC8667] describes the IS-IS extensions that need to be introduced for Segment Routing operating on an MPLS data plane. It defined Adjacency Segment Identifier (Adj-SID) sub-TLV advertised with TLV-22/222/23/223/141, and Adjacency Segment Identifier (LAN-Adj-SID) Sub-TLV advetised with TLV-22/222/23/223. Accordingly, this document defines two new optional Sub-TLVs, "ISIS Adjacency Segment Identifier (Adj-SID) per Algorithm Sub-TLV" and "ISIS Adjacency Segment Identifier (LAN-Adj-SID) per Algorithm Sub-TLV".

4.1.1. ISIS Adjacency Segment Identifier (Adj-SID) per Algorithm Sub-TLV

ISIS Adjacency Segment Identifier (Adj-SID) per Algorithm Sub-TLV has the following format:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |     Flags     |     Weight    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Algorithm   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID/Label/Index (variable)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: ISIS Adjacency Segment Identifier (Adj-SID) per Algorithm Format

where:

Type: TBA1.

Length: 6 or 7 depending on size of the SID.

Flags: Refer to Adjacency Segment Identifier (Adj-SID) sub-TLV.

Weight: Refer to Adjacency Segment Identifier (Adj-SID) sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

SID/Label/Index: Refer to Adjacency Segment Identifier (Adj-SID) sub-TLV.

For a P2P link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

4.1.2. ISIS Adjacency Segment Identifier (LAN-Adj-SID) per Algorithm Sub-TLV

ISIS Adjacency Segment Identifier (LAN-Adj-SID) per Algorithm Sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |     Flags     |    Weight     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Algorithm   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor System-ID (ID length octets)        |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ISIS Adjacency Segment Identifier (LAN-Adj-SID) per Algorithm Format

where:

Type: TBA2.

Length: Variable.

Flags: Refer to Adjacency Segment Identifier (LAN-Adj-SID) Sub-TLV.

Weight: Refer to Adjacency Segment Identifier (LAN-Adj-SID) Sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

SID/Label/Index: Refer to Adjacency Segment Identifier (LAN-Adj-SID) Sub-TLV.

For a broadcast link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

4.2. OSPFv2 Adjacency Segment Identifier per Algorithm

[RFC8665] describes the OSPFv2 extensions that need to be introduced for Segment Routing operating on an MPLS data plane. It defined Adj-SID Sub-TLV and LAN Adj-SID Sub-TLV advertised with Extended Link TLV defined in [RFC7684]. Accordingly, this document defines two new optional Sub-TLVs, "OSPFv2 Adj-SID per Algorithm Sub-TLV" and "OSPFv2 LAN Adj-SID per Algorithm Sub-TLV".

4.2.1. OSPFv2 Adj-SID per Algorithm Sub-TLV

OSPFv2 Adj-SID per Algorithm Sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |   Algorithm   |     MT-ID     |  Weight       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: OSPFv2 Adj-SID per Algorithm Format

where:

Type: TBA3

Length: 7 or 8 octets, depending on the V-Flag.

Flags: Refer to OSPFv2 Adj-SID Sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

MT-ID: Refer to OSPFv2 Adj-SID Sub-TLV.

Weight: Refer to OSPFv2 Adj-SID Sub-TLV.

SID/Index/Label: Refer to OSPFv2 Adj-SID Sub-TLV.

For a P2P link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

4.2.2. OSPFv2 LAN Adj-SID per Algorithm Sub-TLV

OSPFv2 LAN Adj-SID per Algorithm Sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |   Algorithm   |     MT-ID     |    Weight     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Neighbor ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID/Label/Index (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: OSPFv2 LAN Adj-SID per Algorithm Format

where:

Type: TBA4

Length: 11 or 12 octets, depending on the V-Flag.

Flags: Refer to OSPFv2 LAN Adj-SID Sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

MT-ID: Refer to OSPFv2 LAN Adj-SID Sub-TLV.

Weight: Refer to OSPFv2 LAN Adj-SID Sub-TLV.

Neighbor ID: Refer to OSPFv2 LAN Adj-SID Sub-TLV.

SID/Index/Label: Refer to OSPFv2 LAN Adj-SID Sub-TLV.

For a broadcast link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

4.3. OSPFv3 Adjacency Segment Identifier per Algorithm

[RFC8666] describes the OSPFv3 extensions that need to be introduced for Segment Routing operating on an MPLS data plane. It defined Adj-SID Sub-TLV and LAN Adj-SID Sub-TLV advertised with Router-Link TLV as defined in [RFC8362]. Accordingly, this document defines two new optional Sub-TLVs, "OSPFv3 Adj-SID per Algorithm Sub-TLV" and "OSPFv3 LAN Adj-SID per Algorithm Sub-TLV".

4.3.1. OSPFv3 Adj-SID per Algorithm Sub-TLV

OSPFv3 Adj-SID per Algorithm Sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |     Weight    |   Algorithm   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: OSPFv3 Adj-SID per Algorithm Format

where:

Type: TBA5

Length: 7 or 8 octets, depending on the V-Flag.

Flags: Refer to OSPFv3 Adj-SID Sub-TLV.

Weight: Refer to OSPFv3 Adj-SID Sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

Reserved: SHOULD be set to 0 on transmission and MUST be ignored on reception.

SID/Index/Label: Refer to OSPFv3 Adj-SID Sub-TLV.

For a P2P link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

4.3.2. OSPFv3 LAN Adj-SID per Algorithm Sub-TLV

OSPFv3 LAN Adj-SID per Algorithm Sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |     Weight    |   Algorithm   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Neighbor ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID/Label/Index (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: OSPFv3 LAN Adj-SID per Algorithm Format

where:

Type: TBA6

Length: 11 or 12 octets, depending on the V-Flag.

Flags: Refer to OSPFv3 LAN Adj-SID Sub-TLV.

Weight: Refer to OSPFv3 LAN Adj-SID Sub-TLV.

Algorithm: The Algorithm field contains the identifier of the algorithm the router uses to apply algorithm specific treatment configured on the adjacency.

Reserved: SHOULD be set to 0 on transmission and MUST be ignored on reception.

Neighbor ID: Refer to OSPFv3 LAN Adj-SID Sub-TLV.

SID/Index/Label: Refer to OSPFv3 LAN Adj-SID Sub-TLV.

For a broadcast link, an SR-capable router MAY allocate different Adj-SID for different algorithm, if this link joins different algorithm related plane.

5. Operations

The method introduced in this document enables the traffic of different flex-algo plane to be distinguished on the same link, so

that these traffic can be applied with different QoS policy per algorithm.

The endpoint of a link shared by multiple flex-algo plane can reserve different queue resources for different algorithms locally, and perform priority based queue scheduling and traffic shaping. This algorithm related reserved information can be advertised to other nodes in the network through some mechanism, therefore it has an impact on the constraint based path calculation of the flex-algo plane. How to allocate algorithm related resouce and advertise it in the network is out the scope of this document.

Depending on the implementation, operators can configure multiple Adacency-SIDs each for different algorithm on the same link. One of the difficulties is that during this configuration phase it is not straightforward for a link to be included in an FA plane, as this can only be determined after all nodes in the network have negotiated the FAD. A simple way is that as long as an IGP instance enable an algorithm for a level/area, all links joined to that level/area should allocate Adjacency-SIDs for that algorithm statically. Another way is to allocate and withdraw Adjacency-SID per algorithm dynamically according to the result of FAD negotiation.

The following figure shows an example of Adjacency-SID per algorithm.

```
[S1]--------[D]--------[S2]
 |           |          |
 |           |          |
 |           |          |
[A]---------[B]--------[C]
```

Figure 7: Flex-algo LFA Path with Adjacency-SID per Algorithm

Suppose that node S1, A, B, D and their inter-connected links belongs to FA-id 128 plane, and S2, B, C, D and their inter-connected links belongs to FA-id 129 plane. The IGP metric of link B-D is 100, and all other links have IGP metric 1. Both FA-id 128 and 129 use IGP default metric type for path calculation. In FA-id 128 plane, from S1 to destination D, the primary path is S1-D, and the TI-LFA backup path is segment list {node(B), adjacency(B-D)}. Similarly, In FA-id 129 plane, from S2 to destination D, the primary path is S2-D, and the TI-LFA backup path is segment list {node(B), adjacency(B-D)}. The above TI-LFA path of FA-id 128 plane can be translated to {node-SID(B)@FA-id128, adjacency-SID(B-D)@FA-id128}, and TI-LFA path of FA-id 129 plane will be translate to {node-SID(B)@FA-id129, adjacency-SID(B-D)@FA-id129}. So that node B can distinguish the flow of FA-id 128 and FA-id 129 based on different adjacency-SID(B-D), and take

different treatment (e.g., QoS policy) of them when they are send to the same outgoing link B-D.

6. IANA Considerations

6.1. IANA ISIS Considerations

This document makes the following registrations in the "Sub-TLVs for TLV 22, 23, 25, 141, 222, and 223" registry.

Type	Description	22	23	25	141	222	223
TBA1	Adjacency Segment Identifier per Algorithm	y	y	n	y	y	y
TBA2	LAN Adjacency Segment Identifier per Algorithm	y	y	n	y	y	y

6.2. IANA OSPFv2 Considerations

This document makes the following registrations in the OSPFv2 Extended Link TLV Sub-TLVs Registry.

Value	Description	Reference
TBA3	OSPFv2 Adj-SID per Algorithm Sub-TLV	This document
TBA4	OSPFv2 LAN Adj-SID per Algorithm Sub-TLV	This document

6.3. IANA OSPFv3 Considerations

This document makes the following registrations in the "OSPFv3 Extended-LSA Sub-TLVs" Registry.

Value	Description	Reference
TBA5	OSPFv3 Adj-SID per Algorithm Sub-TLV	This document
TBA6	OSPFv3 LAN Adj-SID per Algorithm Sub-TLV	This document

7. Security Considerations

There are no new security issues introduced by the extensions in this document. Refer to [RFC8665], [RFC8666], [RFC8667] for other security considerations.

8. Acknowledgements

TBD

9. Normative References

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.

[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>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[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>.

[RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <https://www.rfc-editor.org/info/rfc8362>.

[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>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

Authors' Addresses

Shaofu Peng
ZTE Corporation
China

Email: peng.shaofu@zte.com.cn

Ran Chen
ZTE Corporation
China

Email: chen.ran@zte.com.cn

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Peter Psenak
Cisco Systems
Slovakia

Email: ppsenak@cisco.com