Network Working Group A. Lindem
Internet-Draft Cisco Systems
Intended status: Standards Track K. Patel
Expires: March 11, 2018 Arrcus
S. Zandi
LinkedIn
R. Raszuk
Bloomberg LP
September 7, 2017

OSPF Extensions for Advertising/Signaling BGP Route Reflector Information
draft-acee-ospf-bgp-rr-01.txt

Abstract

This document specifies an OSPF Router Information (RI) TLV to advertise the BGP Router Reflector capability and peering information. This information can be used by BGP Router Reflector clients to dynamically learn and establish sessions with BGP Router Reflectors in the routing domain.

Table of Contents

1. Introduction

This document specifies an OSPF Router Information (RI) TLV [OSPF-RI] to advertise the BGP Router Reflector [BGP-RR] capability and peering information. This information can be used by BGP Router Reflector clients to dynamically learn and establish sessions with BGP Router Reflectors in the routing domain.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-KEYWORDS].

2. OSPF BGP Route Reflector TLV

The BGP Router Reflector TLV can be used to advertise the route reflector capability, local AS number, BGP peering address, and supported AFI/SAFI pairs using an OSPFv2 [OSPF] or OSPFv3 [OSPFV3] router using the OSPF Router Information LSA [OSPF-RI]. The OSPF Router Information LSA can be advertised in either area or AS scoped RI LSAs. The BGP Router Reflector TLV consists of the following fields:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local AS                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Address Family|            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv4/IPv6 Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv6 Address                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv6 Address                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv6 Address                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          AFI                  |   SAFI          |   AFI       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          AFI  |   SAFI        |   o o o
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length — The length will be 12 for IPv4 peering addresses or 24 for IPv6 peering addresses plus 3 * the number of AFI/SAFI pairs.

Local AS — The Router-Reflector’s local AS number. This can either be used for AS match checking or certain situations where the client’s AS doesn’t match the route reflectors.

Address Family — IANA Address family (1 for IPv4 or 2 for IPv6)

Address — Local IPv4 or IPv6 Address used for BGP peering.

AFI/SAFI — Address Family Identifier (AFI)/ Subsequent Address Family Identifier (SAFI). The AFI/SAFI tuple 0/0 will act as a wildcard indicating all configured AFI/SAFIs.

OSPF BGP Route-Reflector TLV

- The BGP Route Reflector (RR) TLV MAY be advertised multiple times with different peering addresses and AFI/SAFI pairs and MAY be advertised in multiple OSPF RI LSAs. The AFI/SAFI tuple, (0,0) will serve as a wildcard indicating all configured AFI/SAFI tuples. This can be used in deployments where the AF deployment is fairly homogeneous.

- o If different peering addresses are advertised for the same AFI/SAFI pair, the decision of whether a BGP client establishes sessions with one or more of the advertised peering addresses is beyond the scope of this document.

- o If the BGP Router Reflector (RR) TLV has an invalid length or is otherwise malformed, it will not be used for BGP client session establishment. The occurrence of a malformed TLV SHOULD be logged.

3. OSPF Router Information (RI) Opaque LSAs

The OSPF BGP TLV may optionally be advertised in an area-scoped or AS-scoped OSPFv2 Router Information (RI) opaque LSA or OSPFv3 Router Information (RI) LSA [OSPF-RI]. BGP clients may then use the peering address to establish BGP sessions with the advertising route-reflector.

4. Security Considerations

Security considerations for the base OSPF protocol are covered in [OSPF] and [OSPFV3].

5. IANA Considerations

The document will require the following IANA actions:

1. A Router Information TLV type for the BGP Router Reflector TLV will be allocated from the OSPF Router Information (RI) TLVs registry.

6. References

6.1. Normative References

[OSPF] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[OSPF-RI] Lindem, A., Shen, N., Vasseur, J., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, January 2016.

[OSPFV3] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

[RFC-KEYWORDS]
Bradner, S., "Key words for use in RFC's to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[BGP-RR] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

Appendix A. Acknowledgments

The RFC text was produced using Marshall Rose's xml2rfc tool.

Authors' Addresses

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

Email: acee@cisco.com

Keyur Patel
Arrcus

Email: keyur@arrcus.com

Shawn Zandi
LinkedIn

Email: szandi@linkedin.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY 10022
USA

Email: robert@raszuk.net

OSPF Working Group H. Chen
Internet-Draft Huawei Technologies
Intended status: Standards Track M. Toy
Expires: March 15, 2018 Verizon
X. Liu
Jabil
L. Liu
Fujitsu
Z. Li
China Mobile
Y. Yang
Sockrate
September 11, 2017

OSPF Extensions for Broadcast Inter-AS TE Link
draft-chen-ospf-ias-lk-01

Abstract

This document presents extensions to the Open Shortest Path First (OSPF) for advertising broadcast inter-AS Traffic Engineering (TE) links.

Table of Contents

1. Introduction

Connections among different Autonomous Systems (ASes) may be point-to-point (P2P) links and broadcast links. For a P2P inter-AS TE link, RFC 5392 defines a new Opaque LSA, the Inter-AS-TE-v2 LSA, for advertising the OSPFv2 link; and a new OSPFv3 LS type, Inter-AS-TE-v3 LSA, for advertising the OSPFv3 link.

Both the Inter-AS-TE-v2 LSA and Inter-AS-TE-v3 LSA contain one top level TLV:

2 - Link TLV

The Link TLV describes a single link and includes a set of sub-TLVs.

The Link ID sub-TLV defined in RFC 3630 MUST NOT be used in the Link TLV of an Inter-AS-TE-v2 LSA, and the Neighbor ID sub-TLV defined in RFC 5329 MUST NOT be used in the Link TLV of an Inter-AS-TE-v3 LSA.

Instead, the remote ASBR is identified by the inclusion of Remote AS Number sub-TLV and IPv4/IPv6 Remote ASBR ID sub-TLV, which is defined in RFC 5392.

For a P2P inter-AS link, the information about its remote ASBR for replacing its link ID may be configured. For a broadcast inter-AS link, its link ID is the interface IP address of the designated router (DR) of the link in OSPF. Since no OSPF runs over any broadcast inter-AS link, no DR or backup DR (BDR) is selected. It is hard to configure a replacement for DR and BDR.

This document presents extensions to OSPF for advertising broadcast inter-AS TE links through defining a new sub-TLV for a broadcast link without configuring any replacement for DR and BDR on the link.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Information on Inter-AS TE Link

For a broadcast link connecting multiple ASBRs in a number of ASes, on each of the ASBRs X, the following information about the link may be obtained:

1) Link Type: Multi-access
2) Local IP address with mask length
3) Traffic engineering metric
4) Maximum bandwidth
5) Maximum reservable bandwidth
6) Unreserved bandwidth
7) Administrative group
8) SRLG

No remote IP address or link ID (i.e., DR’s interface address) may be obtained.

4. Extensions to OSPF

4.1. sub-TLVs

Two new sub-TLVs are defined. One is for local IPv4 address with mask length; and the other is for local IPv6 address with mask length.

The format of the sub-TLV for a local IPv4 address with mask length is shown as follows.

```
 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 (stTBD1)         |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address (4 bytes)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mask Length  |
+-+-+-+-+-+-+-+-+
```

The IPv4 Address indicates the local IPv4 address of a link. The Mask Length indicates the length of the IPv4 address mask.

The format of the sub-TLV for a local IPv6 address with mask length is illustrated below.

```
 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 (stTBD2)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv6 Address (16 bytes)                   |
~                                                               ~

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mask Length  |
+-+-+-+-+-+-+-+-+
```

The IPv6 Address indicates the local IPv6 address of a link. The Mask Length indicates the length of the IPv6 address mask.

4.2. Procedures

4.2.1. OSPF Router Procedure

For a broadcast inter-AS link connecting to multiple ASBRs, each of the ASBRs as an OSPF router advertises an LSA (Inter-AS-TE-v2 LSA for OSPFv2 or Inter-AS-TE-v3 LSA for OSPFv3) with a link TLV containing sub-TLVs for the information such as 1) 10 8) on the broadcast link described in Section 3.

When TE is enabled on an inter-AS link and the link is up, the ASBR SHOULD advertise this link using the normal procedures for OSPF-TE. When either the link is down or TE is disabled on the link, the ASBR SHOULD withdraw the advertisement. When there are changes to the TE parameters for the link (for example, when the available bandwidth changes), the ASBR SHOULD re-advertise the link but MUST take precautions against excessive re-advertisements.

4.2.2. Super Node Procedure

Suppose that there is a super node, which just receives LSAs from each of ASes (or domains) through a passive OSPF adjacency between the super node and an ASBR or ABR in the AS or domain.

For a new broadcast link connecting multiple routers with no link ID configured, when the super node receives an LSA containing the link attached to router X, it stores the link from X into its TED. It finds the link’s remote end P using the link’s local IP address with network mask. P is a Pseudo node identified by the local IP address of the DR selected from the routers connected to the link. After finding P, it associates the link attached to X with P and the link connected to P with X. If P is not found, a new Pseudo node P is created. The super node associates the link attached to X with P and the link attached to P with X. This creates a bidirectional connection between X and P.

The first router and second router from which the super node receives an LSA containing the link are selected as the DR and BDR respectively. After the DR is down, the BDR node becomes the DR and the router other than the DR with the largest (or smallest) local IP address connecting to the link is selected as the BDR.

When the old DR is down and the BDR becomes the new DR, the super node updates its TED through removing the link between each of routers X and old P (the Pseudo node corresponding to the old DR) and adding a link between each of routers X (still connecting to the broadcast link) and new P (the Pseudo node corresponding to the new DR).

5. Security Considerations

The mechanism described in this document does not raise any new security issues for the OSPF protocols.

6. IANA Considerations

This section specifies requests for IANA allocation.

7. Acknowledgement

The authors would like to thank all for their valuable comments on this draft.

8. References

8.1. 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>.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392, January 2009, <https://www.rfc-editor.org/info/rfc5392>.

[RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <https://www.rfc-editor.org/info/rfc5329>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

8.2. Informative References

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA,
USA

EMail: Huaimo.chen@huawei.com

Mehmet Toy
Verizon
USA

EMail: mehmet.toy@verizon.com

Xufeng Liu
Jabil
McLean, VA
USA

EMail: Xufeng_Liu@jabil.com

Lei Liu
Fujitsu
USA

EMail: lliu@us.fujitsu.com

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

EMail: li_zhenqiang@hotmail.com

Yi Yang
Sockrate
NC
USA

EMail: yyang1998@gmail.com

Open Shortest Path First IGP   P. Psenak, Ed.
Internet-Draft   S. Previdi, Ed.
Intended status: Standards Track   C. Filsfils
Expires: March 9, 2018   Cisco Systems, Inc.
                                                                  H. Gredler
                                                                RtBrick Inc.
                                                                   R. Shakir
                                                                Google, Inc.
                                                              W. Henderickx
                                                                       Nokia
                                                                 J. Tantsura
                                                                  Individual
                                                           September 5, 2017

OSPFv3 Extensions for Segment Routing
draft-ietf-ospf-ospfv3-segment-routing-extensions-10

Abstract

Segment Routing (SR) allows for a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF).

This draft describes the OSPFv3 extensions that are required for Segment Routing.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft will expire on March 9, 2018.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) allows for a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF). Prefix segments represent an ecmp-aware shortest-path to a prefix (or a node), as per the state of the IGP topology. Adjacency segments represent a hop over a specific adjacency between two nodes in the IGP. A prefix segment is typically a multi-hop path while an adjacency segment, in most of the cases, is a one-hop path. SR's control-plane can be applied to both IPv6 and MPLS data-planes, and does not require any additional signaling (other than the regular IGP). For example, when used in MPLS networks, SR paths do not require any LDP or RSVP-TE signaling. Still, SR can interoperate in the presence of LSPs established with RSVP or LDP.

This draft describes the OSPFv3 extensions required for segment routing.

Segment Routing architecture is described in [I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases].

2. Segment Routing Identifiers

Segment Routing defines various types of Segment Identifiers (SIDs): Prefix-SID, Adjacency-SID, LAN Adjacency SID and Binding SID.

2.1. SID/Label Sub-TLV

The SID/Label Sub-TLV appears in multiple TLVs or Sub-TLVs defined later in this document. It is used to advertise the SID or label associated with a prefix or adjacency. The SID/Label TLV has 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SID/Label (variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 3

Length: variable, 3 or 4 bytes

SID/Label: if length is set to 3, then the 20 rightmost bits represent a label. If length is set to 4, then the value represents a 32 bit SID.

The receiving router MUST ignore the SID/Label Sub-TLV if the length is other then 3 or 4.

3. Segment Routing Capabilities

Segment Routing requires some additional capabilities of the router to be advertised to other routers in the area.

These SR capabilities are advertised in OSPFv3 Router Information LSA (defined in [RFC4970]).

3.1. SR-Algorithm TLV

The SR-Algorithm TLV is a TLV of the OSPFv3 Router Information LSA (defined in [RFC4970]).

The SR-Algorithm TLV is optional. It MAY only be advertised once in the OSPFv3 Router Information LSA. If the SID/Label Range TLV, as defined in Section 3.2, is advertised, then the SR-Algorithm TLV MUST also be advertised. If the SR-Algorithm TLV is not advertised by the node, such node is considered as not being segment routing capable.

An OSPFv3 router may use various algorithms when calculating reachability to other nodes in area or to prefixes attached to these nodes. Examples of these algorithms are metric based Shortest Path First (SPF), various sorts of Constrained SPF, etc. The SR-Algorithm TLV allows a router to advertise the algorithms that the router is currently using to other routers in an area. The SR-Algorithm TLV has following structure:

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Algorithm 1  |  Algorithm... |  Algorithm n |                |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: TBD, suggested value 8

Length: variable

Algorithm: Single octet identifying the algorithm. The following value has been defined:

0: Shortest Path First (SPF) algorithm based on link metric. This is the standard shortest path algorithm as computed by the OSPF protocol. Consistent with the deployed practice for link-state protocols, Algorithm 0 permits any node to overwrite the SPF path with a different path based on its local policy. If the SR-Algorithm Sub-TLV is advertised, Algorithm 0 MUST be included.

1: Strict Shortest Path First (SPF) algorithm based on link metric. The algorithm is identical to Algorithm 0 but Algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy at the node claiming the support of Algorithm 1 MUST NOT alter the forwarding decision computed by Algorithm 1.

When multiple SR-Algorithm sub-TLVs are received from a given router the receiver SHOULD use the first occurrence of the sub-TLV in the OSPFv3 Router Information LSA. If the SR-Algorithm sub-TLV appears in multiple OSPFv3 Router Information LSAs that have different flooding scopes, the SR-Algorithm sub-TLV in the OSPFv3 Router Information LSA with the lowest flooding scope SHOULD be used. If the SR-Algorithm sub-TLV appears in multiple OSPFv3 Router Information LSAs that have the same flooding scope, the SR-Algorithm sub-TLV in the OSPFv3 Router Information LSA with the numerically smallest Instance ID SHOULD be used and subsequent instances of the SR-Algorithm sub-TLV SHOULD be ignored.

The RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of the SR-Algorithm TLV propagation, area scope flooding is required.

3.2. SID/Label Range TLV

The SID/Label Range TLV is a TLV of the OSPFv3 Router Information LSA (defined in [RFC4970]).

The SID/Label Sub-TLV MAY appear multiple times and has 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          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Range Size               |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sub-TLVs (variable)                       |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: TBD, suggested value 9

Length: variable

Range Size: 3 octets of SID/label range

Initially, the only supported Sub-TLV is the SID/Label TLV as defined in Section 2.1. The SID/Label advertised in the SID/Label TLV represents the first SID/Label in the advertised range.

Multiple occurrence of the SID/Label Range TLV MAY be advertised, in order to advertise multiple ranges. In such case:

o The originating router MUST encode each range into a different SID/Label Range TLV.

o The originating router decides the order in which the set of SID/Label Range TLVs are advertised in the OSPFv3 Router Information LSA. The originating router MUST ensure the order is same after a graceful restart (using checkpointing, non-volatile storage or any other mechanism) in order to assure the SID/label range and SID index correspondence is preserved across graceful restarts.

o The receiving router must adhere to the order in which the ranges are advertised when calculating a SID/label from the SID index.

o A router not supporting multiple occurrences of the SID/Label Range TLV MUST use first advertised SID/Label Range TLV.

The following example illustrates the advertisement of multiple ranges:

The originating router advertises the following ranges:
   Range 1: [100, 199]
   Range 2: [1000, 1099]
   Range 3: [500, 599]

The receiving routers concatenate the ranges and build the Segment Routing Global Block
(SRGB) is as follows:

```
SRGB = [100, 199]
       [1000, 1099]
       [500, 599]
```

The indexes span multiple ranges:

```
index=0 means label 100
...
index 99 means label 199
index 100 means label 1000
index 199 means label 1099
...
index 200 means label 500
...
```

The RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of the SID/Label Range TLV propagation, area scope flooding is required.

3.3. SR Local Block Sub-TLV

The SR Local Block (SRLB) Sub-TLV contains the range of labels the node has reserved for local SIDs. Local SIDs are used, e.g., for Adjacency-SIDs, and may also be allocated by other components than OSPF protocol. As an example, an application or a controller may instruct the router to allocate a specific local SID. Therefore, in order for such applications or controllers to know what are the local SIDs available in the router, it is required that the router advertises its SRLB. The SRLB Sub-TLV is used for that purpose.

The SR Local Block (SRLB) Sub-TLV is a top-level TLV of the OSPFv3 Router Information Opaque LSA (defined in [RFC7770]).

The SR Local Block Sub-TLV MAY appear multiple times in the OSPFv3 Router Information Opaque LSA and 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Range Size                  |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sub-TLVs (variable)                     |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: TBD, suggested value 12

Length: variable

Range Size: 3 octets of the SID/label range. MUST be higher then 0.

Initially, the only supported Sub-TLV is the SID/Label TLV as defined in Section 2.1. The SID/Label advertised in the SID/Label TLV represents the first SID/Label in the advertised range.

When multiple SRLB sub-TLVs are received from a given router the behavior of the receiving system is undefined.

The originating router MUST NOT advertise overlapping ranges.

Each time a SID from the SRLB is allocated, it SHOULD also be reported to all components (e.g.: controller or applications) in order for these components to have an up-to-date view of the current SRLB allocation. This is required to avoid collision between allocation instructions.

Within the context of OSPFv3, the reporting of local SIDs is done through OSPF Sub-TLVs such as the Adjacency-SID (Section 7). However, the reporting of allocated local SIDs may also be done through other means and protocols which mechanisms are outside the scope of this document.

A router advertising the SRLB TLV may also have other label ranges, outside of the SRLB, used for its local allocation purposes which are NOT advertised in the SRLB. For example, it is possible that an Adjacency-SID is allocated using a local label that is not part of the SRLB.

The OSPFv3 RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of SR Local Block Sub-TLV TLV advertisement, area scope flooding is required.

3.4. SRMS Preference Sub-TLV

The Segment Routing Mapping Server (SRMS) Preference sub-TLV is used to advertise a preference associated with the node that acts as a SR Mapping Server. SRMS preference is defined in [I-D.ietf-spring-conflict-resolution].

The SRMS Preference Sub-TLV is a top-level TLV of the OSPFv3 Router Information Opaque LSA (defined in [RFC7770]).

The SRMS Preference Sub-TLV MAY only be advertised once in the OSPFv3 Router Information Opaque LSA and 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Preference    |                  Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 13

Length: 4 octets

Preference: 1 octet. SRMS preference value from 0 to 255.

When multiple SRMS Preference sub-TLVs are received from a given router the receiver SHOULD use the first occurrence of the sub-TLV in the OSPFv3 Router Information LSA. If the SRMS Preference sub-TLV appears in multiple OSPFv3 Router Information LSAs that have different flooding scopes, the SRLB sub-TLV in the OSPFv3 Router Information LSA with the lowest flooding scope SHOULD be used. If the SRMS Preference sub-TLV appears in multiple OSPFv3 Router Information LSAs that have the same flooding scope, the SRMS

Preference sub-TLV in the OSPFv3 Router Information LSA with the numerically smallest Instance ID SHOULD be used and subsequent instances of the SRMS Preference sub-TLV SHOULD be ignored.

The OSPFv3 RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of the SRMS Preference Sub-TLV advertisement, AS scope flooding is required. If the SRMS advertisements from the SRMS server are only used inside the area to which the SRMS server is attached, area scope flooding may be used.

3.5. SR-Forwarding Capabilities

OSPFv3 router supporting Segment Routing needs to advertise its SR data-plane capabilities. Data-plane capabilities are advertised in OSPF Router Informational Capabilities TLV, which is defined in section 2.3 of RFC 4970 [RFC4970].

Two new bits are allocated in the OSPF Router Informational Capability Bits as follows:

Bit-6 - MPLS IPv6 flag. If set, then the router is capable of processing SR MPLS encapsulated IPv6 packets on all interfaces.

Bit-7 - If set, then the router is capable of processing the IPv6 Segment Routing Header on all interfaces as defined in [I-D.previdi-6man-segment-routing-header].

For the purpose of the SR-Forwarding Capabilities propagation, area scope flooding is required.

4. OSPFv3 Extended Prefix Range TLV

In some cases it is useful to advertise attributes for a range of prefixes. Segment Routing Mapping Server, which is described in [I-D.filsfils-spring-segment-routing-ldp-interop], is an example where we need a single advertisement to advertise SIDs for multiple prefixes from a contiguous address range. The OSPFv3 Extended Prefix Range TLV is defined for this purpose.

The OSPFv3 Extended Prefix Range TLV is a new top level TLV of the following LSAs defined in [I-D.ietf-ospf-ospfv3-lsa-extend]:

E-Intra-Area-Prefix-LSA

E-Inter-Area-Prefix-LSA

E-AS-External-LSA

E-Type-7-LSA

Multiple OSPFv3 Extended Prefix Range TLVs MAY be advertised in these extended LSAs. The OSPFv3 Extended Prefix Range 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |       AF      |          Range Size           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Flags      |                Reserved                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Address Prefix (variable)                    |
|                        ...                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Sub-TLVs (variable)                         |
+-                                                             -+
|                                                               |
```

where:

Type: TBD, suggested value 9.

Length: variable

Prefix length: length of the prefix

AF: 0 - IPv6 unicast

Range size: represents the number of prefixes that are covered by the advertisement. The Range Size MUST NOT exceed the number of prefixes that could be satisfied by the prefix length without including addresses from other than the IPv6 unicast address class.

Flags: 1 octet field. The following flags are defined:

```
  0  1  2  3  4  5  6  7
+--+--+--+--+--+--+--+--+
|IA|  |  |  |  |  |  |  |
+--+--+--+--+--+--+--+--+
```

where:

IA-Flag: Inter-Area flag. If set, advertisement is of inter-area type. ABR that is advertising the OSPF Extended Prefix Range TLV between areas MUST set this bit.

This bit is used to prevent redundant flooding of Prefix Range TLVs between areas as follows:

An ABR always prefers intra-area Prefix Range advertisement over inter-area one.

An ABR does not consider inter-area Prefix Range advertisements coming from non backbone area.

An ABR propagates inter-area Prefix Range advertisement from backbone area to connected non backbone areas only if such advertisement is considered to be the best one.

Address Prefix: the prefix, encoded as an even multiple of 32-bit words, padded with zeroed bits as necessary. This encoding consumes ((PrefixLength + 31) / 32) 32-bit words. The Address Prefix represents the first prefix in the prefix range.

5. Prefix SID Sub-TLV

The Prefix SID Sub-TLV is a Sub-TLV of the following OSPFv3 TLVs as defined in [I-D.ietf-ospf-ospfv3-lsa-extend] and in Section 4:

Intra-Area Prefix TLV

Inter-Area Prefix TLV

External Prefix TLV

OSPFv3 Extended Prefix Range TLV

It MAY appear more than once in the parent TLV and 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    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SID/Index/Label (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 4.

Length: variable

Flags: 1 octet field. The following flags are defined:

```
  0  1  2  3  4  5  6  7
+--+--+--+--+--+--+--+--+
|  |NP|M |E |V |L |  |  |
+--+--+--+--+--+--+--+--+
```

where:

NP-Flag: No-PHP flag. If set, then the penultimate hop MUST NOT pop the Prefix-SID before delivering the packet to the node that advertised the Prefix-SID.

M-Flag: Mapping Server Flag. If set, the SID is advertised from the Segment Routing Mapping Server functionality as described in [I-D.filsfils-spring-segment-routing-ldp-interop].

E-Flag: Explicit-Null Flag. If set, any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with a Prefix-SID having an Explicit-NULL value (0 for IPv4) before forwarding the packet.

The V-Flag: Value/Index Flag. If set, then the Prefix-SID carries an absolute value. If not set, then the Prefix-SID carries an index.

The L-Flag: Local/Global Flag. If set, then the value/index carried by the Prefix-SID has local significance. If not set, then the value/index carried by this Sub-TLV has global significance.

Other bits: Reserved. These MUST be zero when sent and are ignored when received.

Algorithm: one octet identifying the algorithm the Prefix-SID is associated with as defined in Section 3.1.

A router receiving a Prefix-SID from a remote node and with an algorithm value that such remote node has not advertised in the SR-Algorithm sub-TLV (Section 3.1) MUST ignore the Prefix-SID sub-TLV.

SID/Index/Label: label or index value depending on the V-bit setting.

Examples:

A 32 bit global index defining the offset in the SID/Label space advertised by this router - in this case the V and L flags MUST NOT be set.

A 24 bit local label where the 20 rightmost bits are used for encoding the label value - in this case the V and L flags MUST be set.

If multiple Prefix-SIDs are advertised for the same prefix, the receiving router MUST use the first encoded SID and MAY use the subsequent SIDs.

When propagating Prefix-SIDs between areas, if multiple prefix-SIDs are advertised for a prefix, an implementation SHOULD preserve the original order when advertising prefix-SIDs to other areas. This allows implementations that only support a single Prefix-SID to have a consistent view across areas.

When calculating the outgoing label for the prefix, the router MUST take into account E and P flags advertised by the next-hop router, if next-hop router advertised the SID for the prefix. This MUST be done regardless of whether the next-hop router contributes to the best path to the prefix.

The NP-Flag (No-PHP) MUST be set for Prefix-SIDs allocated to inter-area prefixes that are originated by the ABR based on intra-area or inter-area reachability between areas. When the inter-area prefix is generated based on a prefix which is directly attached to the ABR, NP-Flag SHOULD NOT be set

The NP-Flag (No-PHP) MUST be set on the Prefix-SIDs allocated to redistributed prefixes, unless the redistributed prefix is directly attached to ASBR, in which case the NP-Flag SHOULD NOT be set.

If the NP-Flag is not set then any upstream neighbor of the Prefix-SID originator MUST pop the Prefix-SID. This is equivalent to the penultimate hop popping mechanism used in the MPLS dataplane. In such case, MPLS EXP bits of the Prefix-SID are not preserved for the final destination (the Prefix-SID being removed). If the NP-Flag is clear then the received E-flag is ignored.

If the NP-Flag is set then:

If the E-flag is not set then any upstream neighbor of the Prefix-SID originator MUST keep the Prefix-SID on top of the stack. This is useful when the originator of the Prefix-SID must stitch the incoming packet into a continuing MPLS LSP to the final destination. This could occur at an inter-area border router (prefix propagation from one area to another) or at an inter-domain border router (prefix propagation from one domain to another).

If the E-flag is set then any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with a Prefix-SID having an Explicit-NULL value. This is useful, e.g., when the originator of the Prefix-SID is the final destination for the related prefix and the originator wishes to receive the packet with the original EXP bits.

When M-Flag is set, NP-flag and E-flag MUST be ignored at reception.

As the Mapping Server does not specify the originator of a prefix advertisement it is not possible to determine PHP behavior solely based on the Mapping Server advertisement. However, PHP behavior may safely be done in following cases:

Prefix is of intra-area type and the downstream neighbor is the originator of the prefix.

Prefix is of inter-area type and downstream neighbor is an ABR, which is advertising the prefix reachability and is setting LA-bit in the Prefix Options as described in section 3.1 of [I-D.ietf-ospf-ospfv3-lsa-extend].

Prefix is of external type and downstream neighbor is an ASBR, which is advertising the prefix reachability and is setting LA-bit in the Prefix Options as described in section 3.1 of [I-D.ietf-ospf-ospfv3-lsa-extend].

When a Prefix-SID is advertised in an Extended Prefix Range TLV, then the value advertised in Prefix SID Sub-TLV is interpreted as a starting SID value.

Example 1: if the following router addresses (loopback addresses) need to be mapped into the corresponding Prefix SID indexes:

```
Router-A: 192::1/128, Prefix-SID: Index 1
Router-B: 192::2/128, Prefix-SID: Index 2
Router-C: 192::3/128, Prefix-SID: Index 3
Router-D: 192::4/128, Prefix-SID: Index 4
```

then the Address Prefix field in the OSPFv3 Extended Prefix Range TLV is set to 192::1, Prefix Length would be set to 128, Range Size would be set to 4 and the Index value in the Prefix-SID Sub-TLV would be set to 1.

Example 2: If the following prefixes need to be mapped into the corresponding Prefix-SID indexes:

```
10:1:1::0/120,   Prefix-SID: Index 51
10:1:1::100/120, Prefix-SID: Index 52
10:1:1::200/120, Prefix-SID: Index 53
10:1:1::300/120, Prefix-SID: Index 54
10:1:1::400/120, Prefix-SID: Index 55
10:1:1::500/120, Prefix-SID: Index 56
10:1:1::600/120, Prefix-SID: Index 57
```

then the Address Prefix field in the OSPFv3 Extended Prefix Range TLV is set to 10:1:1::0, Prefix Length would be set to 120, Range Size would be set to 7 and the Index value in the Prefix-SID Sub-TLV would be set to 51.

6. SID/Label Binding Sub-TLV

The SID/Label Binding Sub-TLV is used to advertise SID/Label mapping for a path to the prefix.

The SID/Label Binding Sub-TLV MAY be originated by any router in an OSPFv3 domain. The router may advertise a SID/Label binding to a FEC along with at least a single 'nexthop style' anchor. The protocol supports more than one 'nexthop style' anchor to be attached to a SID/Label binding, which results into a simple path description language. In analogy to RSVP the terminology for this is called an 'Explicit Route Object' (ERO). Since ERO style path notation allows anchoring SID/label bindings to both link and node IP addresses, any Label Switched Path (LSP) can be described. Furthermore, SID/Label Bindings from external protocols can also be re-advertised.

The SID/Label Binding Sub-TLV may be used for advertising SID/Label Bindings and their associated Primary and Backup paths. In one single TLV, either a primary ERO Path, backup ERO Path, or both are advertised. If a router wants to advertise multiple parallel paths, then it can generate several TLVs for the same Prefix/FEC. Each occurrence of a Binding TLV for a given FEC Prefix will add a new path.

SID/Label Binding Sub-TLV is a Sub-TLV of the following OSPFv3 TLVs, as defined in [I-D.ietf-ospf-ospfv3-lsa-extend] and in Section 4:

Intra-Area Prefix TLV

Inter-Area Prefix TLV

External Prefix TLV

OSPFv3 Extended Prefix Range TLV

Multiple SID/Label Binding Sub-TLVs can be present in these TLVs. The SID/Label Binding Sub-TLV has 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     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sub-TLVs (variable)                     |
+-                                                             -+
|                                                               |
```

where:

Type: TBD, suggested value 7

Length: variable

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|M|             |
+-+-+-+-+-+-+-+-+
```

where:

M-bit - When the bit is set the binding represents the mirroring context as defined in [I-D.minto-rsvp-lsp-egress-fast-protection].

Weight: weight used for load-balancing purposes. The use of the weight is defined in section 3.5.1 of [I-D.ietf-spring-segment-routing].

SID/Label Binding Sub-TLV currently supports following Sub-TLVs:

SID/Label Sub-TLV as described in Section 2.1. This Sub-TLV MUST appear in the SID/Label Binding Sub-TLV and it MUST only appear once.

ERO Metric Sub-TLV as defined in Section 6.1.

ERO Sub-TLVs as defined in Section 6.2.

6.1. ERO Metric Sub-TLV

The ERO Metric Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The ERO Metric Sub-TLV advertises the cost of an ERO path. It is used to compare the cost of a given source/destination path. A router SHOULD advertise the ERO Metric Sub-TLV in an advertised ERO TLV. The cost of the ERO Metric Sub-TLV SHOULD be set to the cumulative IGP or TE path cost of the advertised ERO. Since manipulation of the Metric field may attract or repel traffic to and from the advertised segment, it MAY be manually overridden.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Metric (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ERO Metric Sub-TLV format

where:

Type: TBD, suggested value 8

Length: Always 4

Metric: A 4 octet metric representing the aggregate IGP or TE path cost.

6.2. ERO Sub-TLVs

All 'ERO' information represents an ordered set which describes the segments of a path. The first ERO Sub-TLV describes the first segment of a path. Similiarly, the last ERO Sub-TLV describes the segment closest to the egress point. If a router extends or stitches a path, it MUST prepend the new segment's path information to the ERO list. This applies equally to advertised backup EROs.

All ERO Sub-TLVs must immediately follow the (SID)/Label Sub-TLV.

All Backup ERO Sub-TLVs must immediately follow the last ERO Sub-TLV.

6.2.1. IPv4 ERO Sub-TLV

IPv4 ERO Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The IPv4 ERO Sub-TLV describes a path segment using IPv4 Address style of encoding. Its semantics have been borrowed from [RFC3209].

```
 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     |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Address (4 octets)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 ERO Sub-TLV format

where:

Type: TBD, suggested value 9

Length: 8 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as 'loose'. Otherwise, the segment path is designated as 'strict'.

IPv4 Address - the address of the explicit route hop.

6.2.2. IPv6 ERO Sub-TLV

IPv6 ERO Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The IPv6 ERO Sub-TLV (Type TBA) describes a path segment using IPv6 Address style of encoding. Its semantics have been borrowed from [RFC3209].

```
 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     |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-                         IPv6 Address                        -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 ERO Sub-TLV format

where:

Type: TBD, suggested value 10

Length: 8 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as 'loose'. Otherwise, the segment path is designated as 'strict'.

IPv6 Address - the address of the explicit route hop.

6.2.3. Unnumbered Interface ID ERO Sub-TLV

The Unnumbered Interface ID ERO Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The appearance and semantics of the 'Unnumbered Interface ID' have been borrowed from [RFC3477].

The Unnumbered Interface-ID ERO Sub-TLV describes a path segment that spans over an unnumbered interface. Unnumbered interfaces are referenced using the interface index. Interface indices are assigned local to the router and therefore not unique within a domain. All elements in an ERO path need to be unique within a domain and hence need to be disambiguated using a domain unique Router-ID.

```
  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     |                   Reserved                    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Router ID                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Interface ID                         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Unnumbered Interface ID ERO Sub-TLV format

Type: TBD, suggested value 11

Length: 12 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as 'loose'. Otherwise, the segment path is designated as 'strict'.

Router-ID: Router-ID of the next-hop.

Interface ID: is the identifier assigned to the link by the router specified by the Router-ID.

6.2.4. IPv4 Backup ERO Sub-TLV

IPv4 Prefix Backup ERO Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The IPv4 Backup ERO Sub-TLV describes a path segment using IPv4 Address style of encoding. Its semantics have been borrowed from [RFC3209].

```
 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         |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Address (4 octets)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Backup ERO Sub-TLV format

where:

Type: TBD, suggested value 12

Length: 8 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as 'loose'. Otherwise, the segment path is designated as 'strict'.'

IPv4 Address - the address of the explicit route hop.

6.2.5. IPv6 Backup ERO Sub-TLV

The IPv6 ERO Sub-TLV is a Sub-TLV of the SID/Label Binding Sub-TLV.

The IPv6 Backup ERO Sub-TLV describes a Backup path segment using IPv6 Address style of encoding. Its appearance and semantics have been borrowed from [RFC3209].

The 'L' bit in the Flags is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

```
 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     |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-                       IPv6 Address                          -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 Backup ERO Sub-TLV format

where:

Type: TBD, suggested value 13

Length: 8 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as ’loose’. Otherwise, the segment path is designated as ’strict’.

IPv6 Address - the address of the explicit route hop.

6.2.6. Unnumbered Interface ID Backup ERO Sub-TLV

The Unnumbered Interface ID Backup Sub-TLV is a Sub-TLV of the SID/ Label Binding Sub-TLV.

The appearance and semantics of the ’Unnumbered Interface ID’ have been borrowed from [RFC3477].

The Unnumbered Interface-ID Backup ERO Sub-TLV describes a path segment that spans over an unnumbered interface. Unnumbered interfaces are referenced using the interface index. Interface indices are assigned local to the router and are therefore not unique within a domain. All elements in an ERO path need to be unique within a domain and hence need to be disambiguated with specification of the unique Router-ID.

```
 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     |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Router ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Unnumbered Interface ID Backup ERO Sub-TLV format

where:

Type: TBD, suggested value 14

Length: 12 bytes

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|L|             |
+-+-+-+-+-+-+-+-+
```

where:

L-bit - If the L-bit is set, then the segment path is designated as 'loose'. Otherwise, the segment path is designated as 'strict'.

Router-ID: Router-ID of the next-hop.

Interface ID: is the identifier assigned to the link by the router specified by the Router-ID.

7. Adjacency Segment Identifier (Adj-SID)

An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

7.1. Adj-SID Sub-TLV

The extended OSPFv3 LSAs, as defined in [I-D.ietf-ospf-ospfv3-lsa-extend], are used to advertise prefix SID in OSPFv3

The Adj-SID Sub-TLV is an optional Sub-TLV of the Router-Link TLV as defined in [I-D.ietf-ospf-ospfv3-lsa-extend]. It MAY appear multiple times in Router-Link TLV. Examples where more than one Adj-SID may be used per neighbor are described in section 4 of [I-D.filsfils-spring-segment-routing-use-cases]. The Adj-SID 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    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+---------------------------------------------------------------+
```

where:

Type: TBD, suggested value 5.

Length: variable.

Flags. 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|V|L|G|P|     |
+-+-+-+-+-+-+-+-+
```

where:

B-Flag: Backup-flag. If set, the Adj-SID refers to an adjacency that is eligible for protection (e.g.: using IPFRR or MPLS-FRR) as described in section 3.5 of [I-D.ietf-spring-segment-routing].

The V-Flag: Value/Index Flag. If set, then the Adj-SID carries an absolute value. If not set, then the Adj-SID carries an index.

The L-Flag: Local/Global Flag. If set, then the value/index carried by the Adj-SID has local significance. If not set, then the value/index carried by this Sub-TLV has global significance.

The G-Flag. Group Flag. When set, the G-Flag indicates that the Adj-SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).

P-Flag. Persistent flag. When set, the P-Flag indicates that the Adj-SID is persistently allocated, i.e., the Adj-SID value remains consistent across router restart and/or interface flap.

Other bits: Reserved. These MUST be zero when sent and are ignored when received.

Weight: weight used for load-balancing purposes. The use of the weight is defined in section 3.5.1 of [I-D.ietf-spring-segment-routing].

SID/Index/Label: label or index value depending on the V-bit setting.

Examples:

A 32 bit global index defining the offset in the SID/Label space advertised by this router - in this case the V and L flags MUST NOT be set.

A 24 bit local label where the 20 rightmost bits are used for encoding the label value - in this case the V and L flags MUST be set.

16 octet IPv6 address - in this case the V-flag MUST be set. The L-flag MUST NOT be set if the IPv6 address is globally unique.

An SR capable router MAY allocate an Adj-SID for each of its adjacencies and set the B-Flag when the adjacency is eligible for protection by an FRR mechanism (IP or MPLS) as described in section 3.5 of [I-D.ietf-spring-segment-routing].

An SR capable router MAY allocate more than one Adj-SID to an adjacency

An SR capable router MAY allocate the same Adj-SID to different adjacencies

When the P-flag is not set, the Adj-SID MAY be persistent. When the P-flag is set, the Adj-SID MUST be persistent.

7.2. LAN Adj-SID Sub-TLV

The LAN Adj-SID is an optional Sub-TLV of the Router-Link TLV. It MAY appear multiple times in the Router-Link TLV. It is used to advertise a SID/Label for an adjacency to a non-DR neighbor on a broadcast or NBMA network.

```
 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    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Neighbor ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID/Label/Index (variable)                 |
+---------------------------------------------------------------+
```

where:

Type: TBD, suggested value 6.

Length: variable.

Flags. 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|V|L|G|P|     |
+-+-+-+-+-+-+-+-+
```

where:

B-Flag: Backup-flag: set if the LAN-Adj-SID refer to an adjacency that is eligible for protection (e.g.: using IPFRR or MPLS-FRR) as described in section 3.1 of [I-D.filsfils-spring-segment-routing-use-cases].

The V-Flag: Value/Index Flag. If set, then the LAN Adj-SID carries an absolute value. If not set, then the LAN Adj-SID carries an index.

The L-Flag: Local/Global Flag. If set, then the value/index carried by the LAN Adj-SID has local significance. If not set, then the value/index carried by this subTLV has global significance.

The G-Flag. Group Flag. When set, the G-Flag indicates that the LAN Adj-SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).

P-Flag. Persistent flag. When set, the P-Flag indicates that the Adj-SID is persistently allocated, i.e., the Adj-SID value remains consistent across router restart and/or interface flap.

Other bits: Reserved. These MUST be zero when sent and are ignored when received.

Weight: weight used for load-balancing purposes. The use of the weight is defined in section 3.5.1 of [I-D.ietf-spring-segment-routing].

Neighbor ID: The Router ID of the neighbor for which the Adj-SID is advertised.

SID/Index/Label: label or index value depending on the V-bit setting.

Examples:

A 32 bit global index defining the offset in the SID/Label space advertised by this router - in this case the V and L flags MUST NOT be set.

A 24 bit local label where the 20 rightmost bits are used for encoding the label value - in this case the V and L flags MUST be set.

16 octet IPv6 address - in this case the V-flag MUST be set. The L-flag MUST NOT be set if the IPv6 address is globally unique.

When the P-flag is not set, the Adj-SID MAY be persistent. When the P-flag is set, the Adj-SID MUST be persistent.

8. Elements of Procedure

8.1. Intra-area Segment routing in OSPFv3

An OSPFv3 router that supports segment routing MAY advertise Prefix-SIDs for any prefix that it is advertising reachability for (e.g., loopback IP address) as described in Section 5.

If multiple routers advertise a Prefix-SID for the same prefix, then the Prefix-SID MUST be the same. This is required in order to allow traffic load-balancing when multiple equal cost paths to the destination exist in the network.

The Prefix-SID can also be advertised by the SR Mapping Servers (as described in [I-D.filsfils-spring-segment-routing-ldp-interop]). The Mapping Server advertises Prefix-SID for remote prefixes that exist in the network. Multiple Mapping Servers can advertise Prefix-SID for the same prefix, in which case the same Prefix-SID MUST be advertised by all of them. The SR Mapping Server could use either area scope or autonomous system flooding scope when advertising Prefix SID for prefixes, based on the configuration of the SR Mapping Server. Depending on the flooding scope used, the SR Mapping Server chooses the LSA that will be used. If the area flooding scope is needed, E-Intra-Area-Prefix-LSA ([I-D.ietf-ospf-ospfv3-lsa-extend]) is used. If autonomous system flooding scope is needed, E-AS-External-LSA ([I-D.ietf-ospf-ospfv3-lsa-extend]) is used.

When a Prefix-SID is advertised by the Mapping Server, which is indicated by the M-flag in the Prefix-SID Sub-TLV (Section 5), the route type as implied by the LSA type is ignored and the Prefix-SID is bound to the corresponding prefix independent of the route type.

Advertisement of the Prefix-SID by the Mapping Server using Inter-Area Prefix TLV, External-Prefix TLV or Intra-Area-Prefix TLV ([I-D.ietf-ospf-ospfv3-lsa-extend]) does not itself contribute to the prefix reachability. The NU-bit MUST be set in the PrefixOptions field of the LSA which is used by the Mapping Server to advertise SID or SID range, which prevents the advertisement to contribute to prefix reachability.

SR Mapping Server MUST use OSPF Extended Prefix Range TLV when advertising SIDs for prefixes. Prefixes of different route-types can be combined in a single OSPF Extended Prefix Range TLV advertised by the SR Mapping Server.

Area scoped OSPF Extended Prefix Range TLV are propagated between areas. Similar to propagation of prefixes between areas, ABR only propagates the OSPF Extended Prefix Range TLV that it considers to be the best from the set it received. The rules used to pick the best OSPF Extended Prefix Range TLV is described in Section 4.

When propagating OSPF Extended Prefix Range TLV between areas, ABR MUST set the IA-Flag, that is used to prevent redundant flooding of the OSPF Extended Prefix Range TLV between areas as described in Section 4.

8.2. Inter-area Segment routing in OSPFv3

In order to support SR in a multi-area environment, OSPFv3 must propagate Prefix-SID information between areas. The following procedure is used in order to propagate Prefix SIDs between areas.

When an OSPFv3 ABR advertises a Inter-Area-Prefix-LSA from an intra-area prefix to all its connected areas, it will also include Prefix-SID Sub-TLV, as described in Section 5. The Prefix-SID value will be set as follows:

The ABR will look at its best path to the prefix in the source area and find out the advertising router associated with the best path to that prefix.

The ABR will then determine if such router advertised a Prefix-SID for the prefix and use it when advertising the Prefix-SID to other connected areas.

If no Prefix-SID was advertised for the prefix in the source area by the router that contributes to the best path to the prefix, the originating ABR will use the Prefix-SID advertised by any other router when propagating Prefix-SID for the prefix to other areas.

When an OSPFv3 ABR advertises Inter-Area-Prefix-LSA LSAs from an inter-area route to all its connected areas it will also include Prefix-SID Sub-TLV, as described in Section 5. The Prefix-SID value will be set as follows:

The ABR will look at its best path to the prefix in the source area and find out the advertising router associated with the best path to that prefix.

The ABR will then look if such router advertised a Prefix-SID for the prefix and use it when advertising the Prefix-SID to other connected areas.

If no Prefix-SID was advertised for the prefix in the source area by the ABR that contributes to the best path to the prefix, the originating ABR will use the Prefix-SID advertised by any other router when propagating Prefix-SID for the prefix to other areas.

8.3. SID for External Prefixes

AS-External-LSAs are flooded domain wide. When an ASBR, which supports SR, generates E-AS-External-LSA, it should also include Prefix-SID Sub-TLV, as described in Section 5. The Prefix-SID value will be set to the SID that has been reserved for that prefix.

When an NSSA ASBR translates an E-NSSA-LSA into an E-AS-External-LSA, it should also advertise the Prefix-SID for the prefix. The NSSA ABR determines its best path to the prefix advertised in the translated E-NSSA-LSA and finds the advertising router associated with that path. If the advertising router has advertised a Prefix-SID for the prefix, then the NSSA ABR uses it when advertising the Prefix-SID in the E-AS-External-LSA. Otherwise the Prefix-SID advertised by any other router will be used.

8.4. Advertisement of Adj-SID

The Adjacency Segment Routing Identifier (Adj-SID) is advertised using the Adj-SID Sub-TLV as described in Section 7.

8.4.1. Advertisement of Adj-SID on Point-to-Point Links

An Adj-SID MAY be advertised for any adjacency on p2p link that is in a state 2-Way or higher. If the adjacency on a p2p link transitions from the FULL state, then the Adj-SID for that adjacency MAY be removed from the area. If the adjacency transitions to a state lower then 2-Way, then the Adj-SID advertisement MUST be removed from the area.

8.4.2. Adjacency SID on Broadcast or NBMA Interfaces

Broadcast or NBMA networks in OSPFv3 are represented by a star topology where the Designated Router (DR) is the central point to which all other routers on the broadcast or NBMA network connect. As a result, routers on the broadcast or NBMA network advertise only their adjacency to the DR. Routers that do not act as DR do not form or advertise adjacencies with each other. They do, however, maintain a 2-Way adjacency state with each other and are directly reachable.

When Segment Routing is used, each router on the broadcast or NBMA network MAY advertise the Adj-SID for its adjacency to the DR using Adj-SID Sub-TLV as described in Section 7.1.

SR capable routers MAY also advertise an Adj-SID for other neighbors (e.g. BDR, DR-OTHER) on the broadcast or NBMA network using the LAN ADJ-SID Sub-TLV as described in Section 7.2.

9. IANA Considerations

This specification updates several existing OSPF registries.

9.1. OSPF Router Information (RI) TLVs Registry

- o 8 (IANA Preallocated) - SR-Algorithm TLV

- o 9 (IANA Preallocated) - SID/Label Range TLV

- o 12 - SR Local Block Sub-TLV

- o 13 - SRMS Preference Sub-TLV

9.2. OSPFv3 Extend-LSA TLV Registry

Following values are allocated:

- o suggested value 9 - OSPF Extended Prefix Range TLV

9.3. OSPFv3 Extend-LSA Sub-TLV registry

- o suggested value 3 - SID/Label Sub-TLV
- o suggested value 4 - Prefix SID Sub-TLV
- o suggested value 5 - Adj-SID Sub-TLV
- o suggested value 6 - LAN Adj-SID Sub-TLV
- o suggested value 7 - SID/Label Binding Sub-TLV
- o suggested value 8 - ERO Metric Sub-TLV
- o suggested value 9 - IPv4 ERO Sub-TLV
- o suggested value 10 - IPv6 ERO Sub-TLV
- o suggested value 11 - Unnumbered Interface ID ERO Sub-TLV
- o suggested value 12 - IPv4 Backup ERO Sub-TLV
- o suggested value 13 - IPv6 Backup ERO Sub-TLV
- o suggested value 14 - Unnumbered Interface ID Backup ERO Sub-TLV

10. Security Considerations

Implementations must assure that malformed permutations of the newly defined sub-TLvs do not result in errors which cause hard OSPFv3 failures.

11. Acknowledgements

Thanks to Acee Lindem for the detail review of the draft, corrections, as well as discussion about details of the encoding.

We would like to thank Anton Smirnov for his contribution.

Many thanks to Yakov Rekhter, John Drake and Shraddha Hedge for their contribution on earlier definition of the "Binding / MPLS Label TLV".

12. References

12.1. 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>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, DOI 10.17487/RFC3477, January 2003, <https://www.rfc-editor.org/info/rfc3477>.

[RFC4970] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, DOI 10.17487/RFC4970, July 2007, <https://www.rfc-editor.org/info/rfc4970>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

12.2. Informative References

[I-D.filsfils-spring-segment-routing-ldp-interop]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing interoperability with LDP", draft-filsfils-spring-segment-routing-ldp-interop-02 (work in progress), September 2014.

[I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.

[I-D.ietf-ospf-ospfv3-lsa-extend]
Lindem, A., Roy, A., Goethals, D., Vallem, V., and F. Baker, "OSPFv3 LSA Extendibility", draft-ietf-ospf-ospfv3-lsa-extend-14 (work in progress), April 2017.

[I-D.ietf-spring-conflict-resolution]
Ginsberg, L., Psenak, P., Previdi, S., and M. Pilka, "Segment Routing Conflict Resolution", draft-ietf-spring-conflict-resolution-01 (work in progress), June 2016.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[I-D.minto-rsvp-lsp-egress-fast-protection]
Jeganathan, J., Gredler, H., and Y. Shen, "RSVP-TE LSP egress fast-protection", draft-minto-rsvp-lsp-egress-fast-protection-03 (work in progress), November 2013.

[I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., Leung, I., Linkova, J., Kosugi, T., Vyncke, E., and D. Lebrun, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-08 (work in progress), October 2015.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems, Inc.
Apollo Business Center
Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Hannes Gredler
RtBrick Inc.
Austria

Email: hannes@rtbrick.com

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: robjs@google.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
BE

Email: wim.henderickx@nokia.com

Jeff Tantsura
Individual
US

Email: jefftant.ietf@gmail.com

Open Shortest Path First IGP   P. Psenak, Ed.
Internet-Draft   S. Previdi, Ed.
Intended status: Standards Track   C. Filsfils
Expires: April 27, 2018   Cisco Systems, Inc.
                                                                H. Gredler
                                                              RtBrick Inc.
                                                                 R. Shakir
                                                              Google, Inc.
                                                           W. Henderickx
                                                                     Nokia
                                                               J. Tantsura
                                                                Individual
                                                          October 24, 2017

OSPF Extensions for Segment Routing
draft-ietf-ospf-segment-routing-extensions-21

Abstract

Segment Routing (SR) allows a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF).

This draft describes the OSPF extensions required for Segment Routing.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft will expire on April 27, 2018.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) allows a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF). Prefix segments represent an ECMP-aware shortest-path to a prefix (or a node), as per the state of the IGP topology. Adjacency segments represent a hop over a specific adjacency between two nodes in the IGP. A prefix segment is typically a multi-hop path while an adjacency segment, in most cases, is a one-hop path. SR's control-plane can be applied to both IPv6 and MPLS data-planes, and does not require any additional signalling (other than IGP extensions). The IPv6 data plane is out of the scope of this specification - it is not applicable to OSPFv2 which only supports the IPv4 address-family. For example, when used in MPLS networks, SR paths do not require any LDP or RSVP-TE signalling. However, SR can interoperate in the presence of LSPs established with RSVP or LDP.

There are additional segment types, e.g., Binding SID defined in [I-D.ietf-spring-segment-routing].

This draft describes the OSPF extensions required for Segment Routing.

Segment Routing architecture is described in [I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in [RFC7855].

2. Segment Routing Identifiers

Segment Routing defines various types of Segment Identifiers (SIDs): Prefix-SID, Adjacency-SID, LAN Adjacency SID, and Binding SID.

Extended Prefix/Link Opaque LSAs defined in [RFC7684] are used for advertisements of the various SID types.

2.1. SID/Label Sub-TLV

The SID/Label Sub-TLV appears in multiple TLVs or Sub-TLVs defined later in this document. It is used to advertise the SID or label associated with a prefix or adjacency. The SID/Label Sub-TLV has 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID/Label (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1

Length: Variable, 3 or 4 octet

SID/Label: If length is set to 3, then the 20 rightmost bits represent a label. If length is set to 4, then the value represents a 32-bit SID.

The receiving router MUST ignore the SID/Label Sub-TLV if the length is other then 3 or 4.

3. Segment Routing Capabilities

Segment Routing requires some additional router capabilities to be advertised to other routers in the area.

These SR capabilities are advertised in the Router Information Opaque LSA (defined in [RFC7770]).

3.1. SR-Algorithm TLV

The SR-Algorithm TLV is a top-level TLV of the Router Information Opaque LSA (defined in [RFC7770]).

The SR-Algorithm TLV is optional. It SHOULD only be advertised once in the Router Information Opaque LSA. If the SR-Algorithm TLV is not advertised by the node, such node is considered as not being segment routing capable.

An SR Router may use various algorithms when calculating reachability to OSPF routers or prefixes in an OSPF area. Examples of these algorithms are metric based Shortest Path First (SPF), various flavors of Constrained SPF, etc. The SR-Algorithm TLV allows a router to advertise the algorithms currently used by the router to other routers in an OSPF area. The SR-Algorithm TLV has 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Algorithm 1 | Algorithm...  |   Algorithm n |               |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: 8

Variable, in octets, dependent on number of algorithms advertised.

Algorithm: Single octet identifying the algorithm. The following values are defined by this document:

0: Shortest Path First (SPF) algorithm based on link metric. This is the standard shortest path algorithm as computed by the OSPF protocol. Consistent with the deployed practice for link-state protocols, Algorithm 0 permits any node to overwrite the SPF path with a different path based on its local policy. If the SR-Algorithm TLV is advertised, Algorithm 0 MUST be included.

1: Strict Shortest Path First (SPF) algorithm based on link metric. The algorithm is identical to Algorithm 0 but Algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy at the node claiming support for Algorithm 1 MUST NOT alter the SPF paths computed by Algorithm 1.

When multiple SR-Algorithm TLVs are received from a given router, the receiver SHOULD use the first occurrence of the TLV in the Router Information LSA. If the SR-Algorithm TLV appears in multiple Router Information LSAs that have different flooding scopes, the SR-Algorithm TLV in the Router Information LSA with the area-scoped flooding scope SHOULD be used. If the SR-Algorithm TLV appears in multiple Router Information LSAs that have the same flooding scope, the SR-Algorithm TLV in the Router Information (RI) LSA with the numerically smallest Instance ID SHOULD be used and subsequent instances of the SR-Algorithm TLV SHOULD be ignored.

The RI LSA can be advertised at any of the defined opaque flooding scopes (link, area, or Autonomous System (AS)). For the purpose of SR-Algorithm TLV advertisement, area-scoped flooding is REQUIRED.

3.2. SID/Label Range TLV

Prefix SIDs MAY be advertised in a form of an index as described in Section 5. Such index defines the offset in the SID/Label space advertised by the router. The SID/Label Range TLV is used to advertise such SID/Label space.

The SID/Label Range TLV is a top-level TLV of the Router Information Opaque LSA (defined in [RFC7770]).

The SID/Label Range TLV MAY appear multiple times and 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Range Size                 |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sub-TLVs (variable)                    |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: 9

Length: Variable, in octets, dependent on Sub-TLVs.

Range Size: 3-octet SID/label range size (i.e., the number of SIDs or labels in the range including the first SID/label). It MUST be greater than 0.

Initially, the only supported Sub-TLV is the SID/Label Sub-TLV as defined in Section 2.1. The SID/Label Sub-TLV MUST be included in the SID/Label Range TLV. The SID/Label advertised in the SID/Label Sub-TLV represents the first SID/Label in the advertised range.

Only a single SID/Label Sub-TLV MAY be advertised in SID/Label Range TLV. If more then one SID/Label Sub-TLVs are present, the SID/Label Range TLV MUST be ignored.

Multiple occurrences of the SID/Label Range TLV MAY be advertised, in order to advertise multiple ranges. In such case:

- o The originating router MUST encode each range into a different SID/Label Range TLV.

- o The originating router decides the order in which the set of SID/Label Range TLVs are advertised inside the Router Information Opaque LSA. The originating router MUST ensure the order is the same after a graceful restart (using checkpointing, non-volatile storage, or any other mechanism) in order to assure the SID/label range and SID index correspondence is preserved across graceful restarts.

- o The receiving router MUST adhere to the order in which the ranges are advertised when calculating a SID/label from a SID index.

- o The originating router MUST NOT advertise overlapping ranges.

- o When a router receives multiple overlapping ranges, it MUST conform to the procedures defined in [I-D.ietf-spring-conflict-resolution].

The following example illustrates the advertisement of multiple ranges:

The originating router advertises the following ranges:

```
Range 1: Range Size: 100   SID/Label Sub-TLV: 100
Range 1: Range Size: 100   SID/Label Sub-TLV: 1000
Range 1: Range Size: 100   SID/Label Sub-TLV: 500
```

The receiving routers concatenate the ranges and build the Segment Routing Global Block (SRGB) as follows:

```
SRGB = [100, 199]
       [1000, 1099]
       [500, 599]
```

The indexes span multiple ranges:

```
index=0 means label 100
...
index 99 means label 199
index 100 means label 1000
index 199 means label 1099
...
index 200 means label 500
...
```

The RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of SID/ Label Range TLV advertisement, area-scoped flooding is REQUIRED.

3.3. SR Local Block TLV

The SR Local Block TLV (SRLB TLV) contains the range of labels the node has reserved for local SIDs. SIDs from the SRLB MAY be used for Adjacency-SIDs, but also by components other than the OSPF protocol. As an example, an application or a controller may instruct the router to allocate a specific local SID. Some controllers or applications may use the control plane to discover the available set of local SIDs on a particular router. In such cases, the SRLG is advertised in the control plane. The requirement to advertise the SRLB is further described in [I-D.ietf-spring-segment-routing-mpls]. The SRLB TLV is used to advertise the SRLB.

The SRLB TLV is a top-level TLV of the Router Information Opaque LSA (defined in [RFC7770]).

The SRLB TLV MAY appear multiple times in the Router Information Opaque LSA and 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Range Size                   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sub-TLVs (variable)                      |
+-                                                             -+
|                                                               |
+                                                               +
```

where:

Type: 14

Length: Variable, in octets, dependent on Sub-TLVs.

Range Size: 3-octet SID/label range size (i.e., the number of SIDs or labels in the range including the first SID/label). It MUST be greater than 0.

Initially, the only supported Sub-TLV is the SID/Label Sub-TLV as defined in Section 2.1. The SID/Label Sub-TLV MUST be included in the SRLB TLV. The SID/Label advertised in the SID/Label Sub-TLV represents the first SID/Label in the advertised range.

Only a single SID/Label Sub-TLV MAY be advertised in the SRLB TLV. If more then one SID/Label Sub-TLVs are present, the SRLB TLV MUST be ignored.

The originating router MUST NOT advertise overlapping ranges.

Each time a SID from the SRLB is allocated, it SHOULD also be reported to all components (e.g., controller or applications) in order for these components to have an up-to-date view of the current SRLB allocation. This is required to avoid collisions between allocation instructions.

Within the context of OSPF, the reporting of local SIDs is done through OSPF Sub-TLVs such as the Adjacency-SID (Section 6). However, the reporting of allocated local SIDs may also be done through other means and protocols which are outside the scope of this document.

A router advertising the SRLB TLV may also have other label ranges, outside of the SRLB, used for its local allocation purposes which are NOT advertised in the SRLB TLV. For example, it is possible that an

Adjacency-SID is allocated using a local label that is not part of the SRLB.

The RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of SRLB TLV advertisement, area-scoped flooding is REQUIRED.

3.4. SRMS Preference TLV

The Segment Routing Mapping Server Preference TLV (SRMS Preference TLV) is used to advertise a preference associated with the node that acts as an SR Mapping Server. The role of an SRMS is described in [I-D.ietf-spring-segment-routing-ldp-interop]. SRMS preference is defined in [I-D.ietf-spring-conflict-resolution].

The SRMS Preference TLV is a top-level TLV of the Router Information Opaque LSA (defined in [RFC7770]).

The SRMS Preference TLV MAY only be advertised once in the Router Information Opaque LSA and 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Preference  |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 13

Length: 4 octets

Preference: 1 octet. SRMS preference value from 0 to 255.

When multiple SRMS Preference TLVs are received from a given router, the receiver SHOULD use the first occurrence of the TLV in the Router Information LSA. If the SRMS Preference TLV appears in multiple Router Information LSAs that have different flooding scopes, the SRMS Preference TLV in the Router Information LSA with the narrowest flooding scope SHOULD be used. If the SRMS Preference TLV appears in multiple Router Information LSAs that have the same flooding scope, the SRMS Preference TLV in the Router Information LSA with the numerically smallest Instance ID SHOULD be used and subsequent instances of the SRMS Preference TLV SHOULD be ignored.

The RI LSA can be advertised at any of the defined flooding scopes (link, area, or autonomous system (AS)). For the purpose of the SRMS Preference TLV advertisement, AS-scoped flooding SHOULD be used. This is because SRMS servers can be located in a different area then consumers of the SRMS advertisements. If the SRMS advertisements from the SRMS server are only used inside the SRMS server's area, area-scoped flooding MAY be used.

4. OSPF Extended Prefix Range TLV

In some cases it is useful to advertise attributes for a range of prefixes. The Segment Routing Mapping Server, which is described in [I-D.ietf-spring-segment-routing-ldp-interop], is an example where we need a single advertisement to advertise SIDs for multiple prefixes from a contiguous address range.

The OSPF Extended Prefix Range TLV, which is a top level TLV of the Extended Prefix LSA described in [RFC7684] is defined for this purpose.

Multiple OSPF Extended Prefix Range TLVs MAY be advertised in each OSPF Extended Prefix Opaque LSA, but all prefix ranges included in a single OSPF Extended Prefix Opaque LSA MUST have the same flooding scope. The OSPF Extended Prefix Range 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |     AF        |         Range Size            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Flags       |                Reserved                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Address Prefix (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sub-TLVs (variable)                      |
+-                                                             -+
|                                                               |
```

where:

Type: 2

Length: Variable, in octets, dependent on Sub-TLVs.

Prefix length: Length of prefix in bits.

AF: Address family for the prefix. Currently, the only supported value is 0 for IPv4 unicast. The inclusion of address family in this TLV allows for future extension.

Range size: Represents the number of prefixes that are covered by the advertisement. The Range Size MUST NOT exceed the number of prefixes that could be satisfied by the prefix length without including the IPv4 multicast address range (224.0.0.0/3).

Flags: Single octet field. The following flags are defined:

```
  0  1  2  3  4  5  6  7
+--+--+--+--+--+--+--+--+
|IA|  |  |  |  |  |  |  |
+--+--+--+--+--+--+--+--+
```

where:

IA-Flag: Inter-Area flag. If set, advertisement is of inter-area type. An ABR that is advertising the OSPF Extended Prefix Range TLV between areas MUST set this bit.

This bit is used to prevent redundant flooding of Prefix Range TLVs between areas as follows:

An ABR only propagates an inter-area Prefix Range advertisement from the backbone area to connected non-backbone areas if the advertisement is considered to be the best one. The following rules are used to select the best range from the set of advertisements for the same Prefix Range:

An ABR always prefers intra-area Prefix Range advertisements over inter-area advertisements.

An ABR does not consider inter-area Prefix Range advertisements coming from non-backbone areas.

Address Prefix: For the address family IPv4 unicast, the prefix itself is encoded as a 32-bit value. The default route is represented by a prefix of length 0. Prefix encoding for other address families is beyond the scope of this specification.

5. Prefix SID Sub-TLV

The Prefix SID Sub-TLV is a Sub-TLV of the OSPF Extended Prefix TLV described in [RFC7684] and the OSPF Extended Prefix Range TLV described in Section 4. It MAY appear more than once in the parent TLV and 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     |   Reserved    |     MT-ID     |    Algorithm  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SID/Index/Label (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 2

Length: 7 or 8 octets, dependent on the V-flag

Flags: Single octet field. The following flags are defined:

```
  0  1  2  3  4  5  6  7
+--+--+--+--+--+--+--+--+
|  |NP|M |E |V |L |  |  |
+--+--+--+--+--+--+--+--+
```

where:

NP-Flag: No-PHP flag. If set, then the penultimate hop MUST NOT pop the Prefix-SID before delivering packets to the node that advertised the Prefix-SID.

M-Flag: Mapping Server Flag. If set, the SID was advertised by a Segment Routing Mapping Server as described in [I-D.ietf-spring-segment-routing-ldp-interop].

E-Flag: Explicit-Null Flag. If set, any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with the Explicit-NULL label (0 for IPv4) before forwarding the packet.

V-Flag: Value/Index Flag. If set, then the Prefix-SID carries an absolute value. If not set, then the Prefix-SID carries an index.

L-Flag: Local/Global Flag. If set, then the value/index carried by the Prefix-SID has local significance. If not set, then the value/index carried by this Sub-TLV has global significance.

Other bits: Reserved. These MUST be zero when sent and are ignored when received.

MT-ID: Multi-Topology ID (as defined in [RFC4915]).

Algorithm: Single octet identifying the algorithm the Prefix-SID is associated with as defined in Section 3.1.

A router receiving a Prefix-SID from a remote node and with an algorithm value that such remote node has not advertised in the SR-Algorithm Sub-TLV (Section 3.1) MUST ignore the Prefix-SID Sub-TLV.

SID/Index/Label: According to the V and L flags, it contains either:

A 32-bit index defining the offset in the SID/Label space advertised by this router.

A 24-bit label where the 20 rightmost bits are used for encoding the label value.

If an OSPF router advertises multiple Prefix-SIDs for the same prefix, topology and algorithm, all of them MUST be ignored.

When calculating the outgoing label for the prefix, the router MUST take into account, as described below, the E, NP and M flags advertised by the next-hop router if that router advertised the SID for the prefix. This MUST be done regardless of whether the next-hop router contributes to the best path to the prefix.

The NP-Flag (No-PHP) MUST be set and the E-flag MUST be clear for Prefix-SIDs allocated to inter-area prefixes that are originated by the ABR based on intra-area or inter-area reachability between areas, unless the advertised prefix is directly attached to the ABR.

The NP-Flag (No-PHP) MUST be set and the E-flag MUST be clear for Prefix-SIDs allocated to redistributed prefixes, unless the redistributed prefix is directly attached to the ASBR.

If the NP-Flag is not set, then any upstream neighbor of the Prefix-SID originator MUST pop the Prefix-SID. This is equivalent to the penultimate hop popping mechanism used in the MPLS dataplane. If the NP-flag is not set, then the received E-flag is ignored.

If the NP-flag is set then:

If the E-flag is not set, then any upstream neighbor of the Prefix-SID originator MUST keep the Prefix-SID on top of the stack. This is useful when the originator of the Prefix-SID must stitch the incoming packet into a continuing MPLS LSP to the final destination. This could occur at an Area Border Router (prefix propagation from one area to another) or at an AS Boundary Router (prefix propagation from one domain to another).

If the E-flag is set, then any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with an Explicit-NULL label. This is useful, e.g., when the originator of the Prefix-SID is the final destination for the related prefix and the originator wishes to receive the packet with the original EXP bits.

When the M-Flag is set, the NP-flag and the E-flag MUST be ignored at reception.

As the Mapping Server does not specify the originator of a prefix advertisement, it is not possible to determine PHP behavior solely based on the Mapping Server advertisement. However, PHP behavior SHOULD be done in following cases:

The Prefix is intra-area type and the downstream neighbor is the originator of the prefix.

The Prefix is inter-area type and downstream neighbor is an ABR, which is advertising prefix reachability and is also generating the Extended Prefix TLV with the A-flag set for this prefix as described in section 2.1 of [RFC7684].

The Prefix is external type and downstream neighbor is an ASBR, which is advertising prefix reachability and is also generating the Extended Prefix TLV with the A-flag set for this prefix as described in section 2.1 of [RFC7684].

When a Prefix-SID is advertised in an Extended Prefix Range TLV, then the value advertised in the Prefix SID Sub-TLV is interpreted as a starting SID/Label value.

Example 1: If the following router addresses (loopback addresses) need to be mapped into the corresponding Prefix SID indexes:

```
Router-A: 192.0.2.1/32, Prefix-SID: Index 1
Router-B: 192.0.2.2/32, Prefix-SID: Index 2
Router-C: 192.0.2.3/32, Prefix-SID: Index 3
Router-D: 192.0.2.4/32, Prefix-SID: Index 4
```

then the Prefix field in the Extended Prefix Range TLV would be set to 192.0.2.1, Prefix Length would be set to 32, Range Size would be set to 4, and the Index value in the Prefix-SID Sub-TLV would be set to 1.

Example 2: If the following prefixes need to be mapped into the corresponding Prefix-SID indexes:

```
192.0.2.0/30, Prefix-SID: Index 51
192.0.2.4/30, Prefix-SID: Index 52
192.0.2.8/30, Prefix-SID: Index 53
192.0.2.12/30, Prefix-SID: Index 54
192.0.2.16/30, Prefix-SID: Index 55
192.0.2.20/30, Prefix-SID: Index 56
192.0.2.24/30, Prefix-SID: Index 57
```

then the Prefix field in the Extended Prefix Range TLV would be set to 192.0.2.0, Prefix Length would be set to 30, Range Size would be 7, and the Index value in the Prefix-SID Sub-TLV would be set to 51.

6. Adjacency Segment Identifier (Adj-SID)

An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

6.1. Adj-SID Sub-TLV

Adj-SID is an optional Sub-TLV of the Extended Link TLV defined in [RFC7684]. It MAY appear multiple times in the Extended Link TLV. The Adj-SID 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     |    Reserved   |   MT-ID       |  Weight       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+---------------------------------------------------------------+
```

where:

Type: 2

Length: 7 or 8 octets, dependent on the V flag.

Flags: Single octet field containing the following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|V|L|G|P|     |
+-+-+-+-+-+-+-+-+
```

where:

B-Flag: Backup Flag. If set, the Adj-SID refers to an adjacency that is eligible for protection (e.g., using IPFRR or MPLS-FRR) as described in section 3.5 of [I-D.ietf-spring-segment-routing].

The V-Flag: Value/Index Flag. If set, then the Adj-SID carries an absolute value. If not set, then the Adj-SID carries an index.

The L-Flag: Local/Global Flag. If set, then the value/index carried by the Adj-SID has local significance. If not set, then the value/index carried by this Sub-TLV has global significance.

The G-Flag: Group Flag. When set, the G-Flag indicates that the Adj-SID refers to a group of adjacencies (and therefore MAY be assigned to other adjacencies as well).

P-Flag. Persistent flag. When set, the P-Flag indicates that the Adj-SID is persistently allocated, i.e., the Adj-SID value remains consistent across router restart and/or interface flap.

Other bits: Reserved. These MUST be zero when sent and are ignored when received.

MT-ID: Multi-Topology ID (as defined in [RFC4915].

Weight: Weight used for load-balancing purposes. The use of the weight is defined in [I-D.ietf-spring-segment-routing].

SID/Index/Label: According to the V and L flags, it contains either:

A 32-bit index defining the offset in the SID/Label space advertised by this router.

A 24-bit label where the 20 rightmost bits are used for encoding the label value.

An SR capable router MAY allocate an Adj-SID for each of its adjacencies and set the B-Flag when the adjacency is eligible for protection by an FRR mechanism (IP or MPLS) as described in section 3.5 of [I-D.ietf-spring-segment-routing].

An SR capable router MAY allocate more than one Adj-SID to an adjacency

An SR capable router MAY allocate the same Adj-SID to different adjacencies

When the P-flag is not set, the Adj-SID MAY be persistent. When the P-flag is set, the Adj-SID MUST be persistent.

6.2. LAN Adj-SID Sub-TLV

LAN Adj-SID is an optional Sub-TLV of the Extended Link TLV defined in [RFC7684]. It MAY appear multiple times in the Extended-Link TLV. It is used to advertise a SID/Label for an adjacency to a non-DR router on a broadcast, NBMA, or hybrid [RFC6845] network.

```
 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     |    Reserved   |     MT-ID     |    Weight     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Neighbor ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID/Label/Index (variable)                 |
+---------------------------------------------------------------+
```

where:

Type: 3

Length: 11 or 12 octets, dependent on V-flag.

Flags: same as in Section 6.1

MT-ID: Multi-Topology ID (as defined in [RFC4915].

Weight: Weight used for load-balancing purposes. The use of the weight is defined in [I-D.ietf-spring-segment-routing].

Neighbor ID: The Router ID of the neighbor for which the LAN-Adj-SID is advertised.

SID/Index/Label: According to the V and L flags, it contains either:

A 32-bit index defining the offset in the SID/Label space advertised by this router.

A 24-bit label where the 20 rightmost bits are used for encoding the label value.

When the P-flag is not set, the Adj-SID MAY be persistent. When the P-flag is set, the Adj-SID MUST be persistent.

7. Elements of Procedure

7.1. Intra-area Segment routing in OSPFv2

An OSPFv2 router that supports segment routing MAY advertise Prefix-SIDs for any prefix to which it is advertising reachability (e.g., a loopback IP address as described in Section 5).

A Prefix-SID can also be advertised by the SR Mapping Servers (as described in [I-D.ietf-spring-segment-routing-ldp-interop]). A Mapping Server advertises Prefix-SIDs for remote prefixes that exist in the OSPFv2 routing domain. Multiple Mapping Servers can advertise Prefix-SIDs for the same prefix, in which case the same Prefix-SID MUST be advertised by all of them. The flooding scope of the OSPF Extended Prefix Opaque LSA that is generated by the SR Mapping Server could be either area-scoped or AS-scoped and is determined based on the configuration of the SR Mapping Server.

An SR Mapping Server MUST use the OSPF Extended Prefix Range TLV when advertising SIDs for prefixes. Prefixes of different route-types can be combined in a single OSPF Extended Prefix Range TLV advertised by an SR Mapping Server. Because the OSPF Extended Prefix Range TLV doesn’t include a Route-Type field, as in the OSPF Extended Prefix TLV, it is possible to include adjacent prefixes from different Route-Types in the OSPF Extended Prefix Range TLV.

Area-scoped OSPF Extended Prefix Range TLVs are propagated between areas. Similar to propagation of prefixes between areas, an ABR only propagates the OSPF Extended Prefix Range TLV that it considers to be the best from the set it received. The rules used to pick the best OSPF Extended Prefix Range TLV are described in Section 4.

When propagating an OSPF Extended Prefix Range TLV between areas, ABRs MUST set the IA-Flag, that is used to prevent redundant flooding of the OSPF Extended Prefix Range TLV between areas as described in Section 4.

7.2. Inter-area Segment routing in OSPFv2

In order to support SR in a multi-area environment, OSPFv2 must propagate Prefix-SID information between areas. The following procedure is used to propagate Prefix SIDs between areas.

When an OSPF ABR advertises a Type-3 Summary LSA from an intra-area prefix to all its connected areas, it will also originate an Extended Prefix Opaque LSA, as described in [RFC7684]. The flooding scope of the Extended Prefix Opaque LSA type will be set to area-local scope. The route-type in the OSPF Extended Prefix TLV is set to inter-area. The Prefix-SID Sub-TLV will be included in this LSA and the Prefix-SID value will be set as follows:

The ABR will look at its best path to the prefix in the source area and find the advertising router associated with the best path to that prefix.

The ABR will then determine if such router advertised a Prefix-SID for the prefix and use it when advertising the Prefix-SID to other connected areas.

If no Prefix-SID was advertised for the prefix in the source area by the router that contributes to the best path to the prefix, the originating ABR will use the Prefix-SID advertised by any other router when propagating the Prefix-SID for the prefix to other areas.

When an OSPF ABR advertises Type-3 Summary LSAs from an inter-area route to all its connected areas, it will also originate an Extended Prefix Opaque LSA, as described in [RFC7684]. The flooding scope of the Extended Prefix Opaque LSA type will be set to area-local scope. The route-type in OSPF Extended Prefix TLV is set to inter-area. The Prefix-SID Sub-TLV will be included in this LSA and the Prefix-SID will be set as follows:

The ABR will look at its best path to the prefix in the backbone area and find the advertising router associated with the best path to that prefix.

The ABR will then determine if such router advertised a Prefix-SID for the prefix and use it when advertising the Prefix-SID to other connected areas.

If no Prefix-SID was advertised for the prefix in the backbone area by the ABR that contributes to the best path to the prefix, the originating ABR will use the Prefix-SID advertised by any other router when propagating the Prefix-SID for the prefix to other areas.

7.3. Segment Routing for External Prefixes

Type-5 LSAs are flooded domain wide. When an ASBR, which supports SR, generates Type-5 LSAs, it should also originate Extended Prefix Opaque LSAs, as described in [RFC7684]. The flooding scope of the Extended Prefix Opaque LSA type is set to AS-wide scope. The route-type in the OSPF Extended Prefix TLV is set to external. The Prefix-SID Sub-TLV is included in this LSA and the Prefix-SID value will be set to the SID that has been reserved for that prefix.

When an NSSA ABR translates Type-7 LSAs into Type-5 LSAs, it should also advertise the Prefix-SID for the prefix. The NSSA ABR determines its best path to the prefix advertised in the translated Type-7 LSA and finds the advertising router associated with that path. If the advertising router has advertised a Prefix-SID for the prefix, then the NSSA ABR uses it when advertising the Prefix-SID for the Type-5 prefix. Otherwise, the Prefix-SID advertised by any other router will be used.

7.4. Advertisement of Adj-SID

The Adjacency Segment Routing Identifier (Adj-SID) is advertised using the Adj-SID Sub-TLV as described in Section 6.

7.4.1. Advertisement of Adj-SID on Point-to-Point Links

An Adj-SID MAY be advertised for any adjacency on a P2P link that is in neighbor state 2-Way or higher. If the adjacency on a P2P link transitions from the FULL state, then the Adj-SID for that adjacency MAY be removed from the area. If the adjacency transitions to a state lower then 2-Way, then the Adj-SID advertisement MUST be withdrawn from the area.

7.4.2. Adjacency SID on Broadcast or NBMA Interfaces

Broadcast, NBMA, or hybrid [RFC6845] networks in OSPF are represented by a star topology where the Designated Router (DR) is the central point to which all other routers on the broadcast, NBMA, or hybrid network connect. As a result, routers on the broadcast, NBMA, or hybrid network advertise only their adjacency to the DR. Routers that do not act as DR do not form or advertise adjacencies with each other. They do, however, maintain 2-Way adjacency state with each other and are directly reachable.

When Segment Routing is used, each router on the broadcast, NBMA, or hybrid network MAY advertise the Adj-SID for its adjacency to the DR using the Adj-SID Sub-TLV as described in Section 6.1.

SR capable routers MAY also advertise a LAN-Adj-SID for other neighbors (e.g., BDR, DR-OTHER) on the broadcast, NBMA, or hybrid network using the LAN-ADJ-SID Sub-TLV as described in Section 6.2.

8. IANA Considerations

This specification updates several existing OSPF registries.

8.1. OSPF OSPF Router Information (RI) TLVs Registry

- o 8 (IANA Preallocated) - SR-Algorithm TLV
- o 9 (IANA Preallocated) - SID/Label Range TLV
- o 13 - SRMS Preference TLV
- o 14 - SR Local Block TLV

8.2. OSPF Extended Prefix LSA TLV Registry

Following values are allocated:

- o 2 - OSPF Extended Prefix Range TLV

8.3. OSPF Extended Prefix LSA Sub-TLV Registry

Following values are allocated:

- o 1 - SID/Label Sub-TLV
- o 2 - Prefix SID Sub-TLV

8.4. OSPF Extended Link LSA Sub-TLV Registry

Following initial values are allocated:

- o 1 - SID/Label Sub-TLV
- o 2 - Adj-SID Sub-TLV
- o 3 - LAN Adj-SID/Label Sub-TLV

8.5. IGP Algorithm Type Registry

IANA is requested to set up a registry called "IGP Algorithm Type". The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

Values in this registry must come from the range 0-255.

The initial values in the IGP Algorithm Type registry are:

0: Shortest Path First (SPF) algorithm based on link metric. This is the standard shortest path algorithm as computed by the IGP protocol. Consistent with the deployed practice for link-state protocols, Algorithm 0 permits any node to overwrite the SPF path with a different path based on its local policy.

1: Strict Shortest Path First (SPF) algorithm based on link metric. The algorithm is identical to Algorithm 0 but Algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy at the node claiming support for Algorithm 1 MUST NOT alter the SPF paths computed by Algorithm 1.

9. Implementation Status

An implementation survey with seven questions related to the implementer’s support of OSPFv2 Segment Routing was sent to the OSPF WG list and several known implementers. This section contains responses from three implementers who completed the survey. No external means were used to verify the accuracy of the information submitted by the respondents. The respondents are considered experts on the products they reported on. Additionally, responses were omitted from implementers who indicated that they have not implemented the function yet.

This section will be removed before publication as an RFC.

Responses from Nokia (former Alcatel-Lucent):

Link to a web page describing the implementation:
https://infoproducts.alcatel-lucent.com/cgi-bin/dbaccessfilename.cgi/
3HE10799AAAATQZZA01_V1_7450%20ESS%207750%20SR%20and%207950%20XRS%20Un
icast%20Routing%20Protocols%20Guide%20R14.0.R1.pdf

The implementation’s level of maturity: Production.

Coverage: We have implemented all sections and have support for the latest draft.

Licensing: Part of the software package that needs to be purchased.

Implementation experience: Great spec. We also performed inter-operability testing with Cisco's OSPF Segment Routing implementation.

Contact information: wim.henderickx@nokia.com

Responses from Cisco Systems:

Link to a web page describing the implementation:

http://www.segment-routing.net/home/tutorial

The implementation's level of maturity: Production.

Coverage: All sections have been implemented according to the latest draft.

Licensing: Part of a commercial software package.

Implementation experience: Many aspects of the draft are result of the actual implementation experience, as the draft evolved from its initial version to the current one. Interoperability testing with Alcatel-Lucent was performed, which confirmed the draft's ability to serve as a reference for the implementors.

Contact information: ppsenak@cisco.com

Responses from Juniper:

The implementation's name and/or a link to a web page describing the implementation:

Feature name is OSPF SPRING

The implementation's level of maturity: To be released in 16.2 (second half of 2016)

Coverage: All sections implemented except Sections 4, and 6.

Licensing: JUNOS Licensing needed.

Implementation experience: NA

Contact information: shraddha@juniper.net

10. Security Considerations

With the OSPFv2 segment routing extensions defined herein, OSPFv2 will now program the MPLS data plane [RFC3031] in addition to the IP data plane. Previously, LDP [RFC5036] or another label distribution mechanism was required to advertise MPLS labels and program the MPLS data plane.

In general, the same types of attacks that can be carried out on the IP control plane can be carried out on the MPLS control plane resulting in traffic being misrouted in the respective data planes. However, the latter may be more difficult to detect and isolate.

Existing security extensions as described in [RFC2328] and [RFC7684] apply to these segment routing extensions. While OSPF is under a single administrative domain, there may be deployments where potential attackers have access to one or more networks in the OSPF routing domain. In these deployments, stronger authentication mechanisms such as those specified in [RFC7474] SHOULD be used.

Implementations must assure that malformed TLV and Sub-TLV defined in this document are detected and do not provide a vulnerability for attackers to crash the OSPFv2 router or routing process. Reception of malformed TLV or Sub-TLV SHOULD be counted and/or logged for further analysis. Logging of malformed TLVs and Sub-TLVs should be rate-limited to prevent a Denial of Service (DoS) attack (distributed or otherwise) from overloading the OSPF control plane.

11. Contributors

The following people gave a substantial contribution to the content of this document: Acee Lindem, Ahmed Bashandy, Martin Horneffer, Bruno Decraene, Stephane Litkowski, Igor Milojevic, Rob Shakir and Saku Ytti.

12. Acknowledgements

We would like to thank Anton Smirnov for his contribution.

Thanks to Acee Lindem for the detail review of the draft, corrections, as well as discussion about details of the encoding.

13. References

13.1. Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-12 (work in progress), June 2017.

[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>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC6845] Sheth, N., Wang, L., and J. Zhang, "OSPF Hybrid Broadcast and Point-to-Multipoint Interface Type", RFC 6845, DOI 10.17487/RFC6845, January 2013, <https://www.rfc-editor.org/info/rfc6845>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/info/rfc7120>.

[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>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

13.2. Informative References

[I-D.ietf-spring-conflict-resolution]
Ginsberg, L., Psenak, P., Previdi, S., and M. Pilka, "Segment Routing MPLS Conflict Resolution", draft-ietf-spring-conflict-resolution-05 (work in progress), July 2017.

[I-D.ietf-spring-segment-routing-ldp-interop]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., and S. Litkowski, "Segment Routing interworking with LDP", draft-ietf-spring-segment-routing-ldp-interop-09 (work in progress), September 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <https://www.rfc-editor.org/info/rfc7474>.

[RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements", RFC 7855, DOI 10.17487/RFC7855, May 2016, <https://www.rfc-editor.org/info/rfc7855>.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems, Inc.
Apollo Business Center
Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: stefano@previdi.net

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: robjs@google.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
BE

Email: wim.henderickx@nokia.com

Jeff Tantsura
Individual
US

Email: jefftant.ietf@gmail.com

Internet D. Yeung
Internet-Draft Arrcus
Intended status: Informational Y. Qu
Expires: January 3, 2018 Huawei
J. Zhang
Juniper Networks
I. Chen
Jabil Circuit
A. Lindem
Cisco Systems
July 2, 2017

Yang Data Model for OSPF SR (Segment Routing) Protocol
draft-ietf-ospf-sr-yang-02

Abstract

This document defines a YANG data model that can be used to configure and manage OSPF Segment Routing.

Status of This Memo

Copyright Notice

Table of Contents

1. Overview

YANG [RFC6020] [RFC7950] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g., ReST) and encodings other than XML (e.g., JSON) are being defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage OSPF Segment Routing and it is an augmentation to the OSPF YANG data model.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. OSPF Segment Routing

This document defines a model for OSPF Segment Routing feature. It is an augmentation of the OSPF base model.

The OSPF SR YANG module requires support for the base segment routing module [I-D.ietf-spring-sr-yang], which defines the global segment routing configuration independent of any specific routing protocol configuration, and support of OSPF base model[I-D.ietf-ospf-yang] which defines basic OSPF configuration and state.

```
module: ietf-ospf-sr
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf/ospf:instance:
    +--rw segment-routing
    |  +--rw enabled?    boolean
    |  +--rw bindings
    |     +--rw advertise
    |     |  +--rw policies*   string
    |     +--rw receive?     boolean
    +--rw protocol-srgb {sr:protocol-srgb}?
       +--rw srgb* [lower-bound upper-bound]
          +--rw lower-bound    uint32
          +--rw upper-bound    uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf/ospf:instance
          /ospf:areas/ospf:area/ospf:interfaces/ospf:interface:
    +--rw segment-routing
       +--rw adjacency-sid
          +--rw advertise-adj-group-sid* [group-id]
          |  +--rw group-id    uint32
          +--rw advertise-protection?      enumeration
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf/ospf:instance
          /ospf:areas/ospf:area/ospf:interfaces/ospf:interface
          /ospf:fast-reroute:
    +--rw ti-lfa {ti-lfa}?
       +--rw enable?   boolean
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf/ospf:instance
          /ospf:areas/ospf:area/ospf:interfaces/ospf:interface
          /ospf:database/ospf:link-scope-lsa-type/ospf:link-scope-lsas
          /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
          /ospf:body/ospf:opaque/ospf:extended-prefix-tlvs
          /ospf:extended-prefix-tlv:
    +--ro perfix-sid-sub-tlvs
       +--ro prefix-sid-sub-tlv*
          +--ro flags?       bits
          +--ro mt-id?       uint8
          +--ro algorithm?   uint8
          +--ro sid?         uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf/ospf:instance
          /ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type
          /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
          /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
```

```
            /ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv:
      +--ro perfix-sid-sub-tlvs
         +--ro prefix-sid-sub-tlv*
            +--ro flags?       bits
            +--ro mt-id?       uint8
            +--ro algorithm?   uint8
            +--ro sid?         uint32
    augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:instance
            /ospf:database/ospf:as-scope-lsa-type/ospf:as-scope-lsas
            /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
            /ospf:body/ospf:opaque/ospf:extended-prefix-tlvs
            /ospf:extended-prefix-tlv:
      +--ro perfix-sid-sub-tlvs
         +--ro prefix-sid-sub-tlv*
            +--ro flags?       bits
            +--ro mt-id?       uint8
            +--ro algorithm?   uint8
            +--ro sid?         uint32
    augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:instance
            /ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type
            /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
            /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
            /ospf:extended-link-tlvs/ospf:extended-link-tlv:
      +--ro adj-sid-sub-tlvs
      |  +--ro adj-sid-sub-tlv*
      |     +--ro flags?    bits
      |     +--ro mt-id?    uint8
      |     +--ro weight?   uint8
      |     +--ro sid?      uint32
      +--ro lan-adj-sid-sub-tlvs
         +--ro lan-adj-sid-sub-tlv*
            +--ro flags?                bits
            +--ro mt-id?                uint8
            +--ro weight?               uint8
            +--ro neighbor-router-id?   yang:dotted-quad
            +--ro sid?                  uint32
    augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:instance
            /ospf:areas/ospf:area/ospf:interfaces/ospf:interface
            /ospf:database/ospf:link-scope-lsa-type/ospf:link-scope-lsas
            /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
            /ospf:body/ospf:opaque:
      +--ro extended-prefix-range-tlvs
      |  +--ro extended-prefix-range-tlv*
      |     +--ro range-size?            uint16
      |     +--ro flags?                 bits
```

```
       |     +--ro prefix?                inet:ip-prefix
       |     +--ro perfix-sid-sub-tlvs
       |     |  +--ro prefix-sid-sub-tlv*
       |     |     +--ro flags?       bits
       |     |     +--ro mt-id?       uint8
       |     |     +--ro algorithm?   uint8
       |     |     +--ro sid?         uint32
       |     +--ro unknown-tlvs
       |        +--ro unknown-tlv*
       |           +--ro type?     uint16
       |           +--ro length?   uint16
       |           +--ro value?    yang:hex-string
       +--ro sr-algorithm-tlv
       |  +--ro sr-algorithm*   uint8
       +--ro sid-range-tlvs
       |  +--ro sid-range-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro local-block-tlvs
       |  +--ro local-block-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro srms-preference-tlv
          +--ro preference?   uint8
     augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:instance
             /ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque:
       +--ro extended-prefix-range-tlvs
       |  +--ro extended-prefix-range-tlv*
       |     +--ro range-size?            uint16
       |     +--ro flags?                 bits
       |     +--ro prefix?                inet:ip-prefix
       |     +--ro perfix-sid-sub-tlvs
       |     |  +--ro prefix-sid-sub-tlv*
       |     |     +--ro flags?       bits
       |     |     +--ro mt-id?       uint8
       |     |     +--ro algorithm?   uint8
       |     |     +--ro sid?         uint32
       |     +--ro unknown-tlvs
       |        +--ro unknown-tlv*
       |           +--ro type?     uint16
       |           +--ro length?   uint16
       |           +--ro value?    yang:hex-string
       +--ro sr-algorithm-tlv
```

```
       |  +--ro sr-algorithm*   uint8
       +--ro sid-range-tlvs
       |  +--ro sid-range-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro local-block-tlvs
       |  +--ro local-block-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro srms-preference-tlv
          +--ro preference?   uint8
     augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:instance
             /ospf:database/ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
             /ospf:body/ospf:opaque:
       +--ro extended-prefix-range-tlvs
       |  +--ro extended-prefix-range-tlv*
       |     +--ro range-size?            uint16
       |     +--ro flags?                 bits
       |     +--ro prefix?                inet:ip-prefix
       |     +--ro perfix-sid-sub-tlvs
       |     |  +--ro prefix-sid-sub-tlv*
       |     |     +--ro flags?       bits
       |     |     +--ro mt-id?       uint8
       |     |     +--ro algorithm?   uint8
       |     |     +--ro sid?         uint32
       |     +--ro unknown-tlvs
       |        +--ro unknown-tlv*
       |           +--ro type?     uint16
       |           +--ro length?   uint16
       |           +--ro value?    yang:hex-string
       +--ro sr-algorithm-tlv
       |  +--ro sr-algorithm*   uint8
       +--ro sid-range-tlvs
       |  +--ro sid-range-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro local-block-tlvs
       |  +--ro local-block-tlv*
       |     +--ro range-size?    ospf:uint24
       |     +--ro sid-sub-tlv
       |        +--ro sid?   uint32
       +--ro srms-preference-tlv
          +--ro preference?   uint8
```

3. OSPF Segment Routing Yang Module

```
<CODE BEGINS> file "ietf-ospf-sr@2017-07-02.yang"
module ietf-ospf-sr {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-sr";

  prefix ospf-sr;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-routing {
    prefix "rt";
  }
  import ietf-segment-routing-common {
    prefix "sr-cmn";
  }
  import ietf-segment-routing {
    prefix "sr";
  }
  import ietf-ospf {
    prefix "ospf";
  }

  organization
    "IETF OSPF - OSPF Working Group";

  contact
     "WG Web:   <http://tools.ietf.org/wg/ospf/>
      WG List:  <mailto:ospf@ietf.org>

      Editor:   Derek Yeung
                <mailto:derek@arrcus.com>
      Author:   Derek Yeung
                <mailto:derek@arrcus.com>
      Author:   Yingzhen Qu
                <mailto:yingzhen.qu@huawei.com>
      Author:   Acee Lindem
                <mailto:acee@cisco.com>
      Author:   Jeffrey Zhang
                <mailto:zzhang@juniper.net>
      Author:   Ing-Wher Chen
                <mailto:ing-wher_chen@jabil.com>
```

```
         Author:   Greg Hankins
                   <mailto:greg.hankins@alcatel-lucent.com>";

     description
       "This YANG module defines the generic configuration
        and operational state for OSPF Segment Routing, which is
        common across all of the vendor implementations. It is
        intended that the module will be extended by vendors to
        define vendor-specific OSPF Segment Routing configuration
        and operational parameters and policies.

        Copyright (c) 2017 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX;
        see the RFC itself for full legal notices.";

     reference "RFC XXXX";

     revision 2017-07-02 {
       description
         "* Implement NMDA model.
          * Add local-block-tlvs and srms-preference-tlv.
          * Remove sid-binding-sub-tlvs.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2017-03-12 {
       description
         "* Add p-flag in adj-sid sub-tlv.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2016-10-31 {
       description
         "* Update authors information.
          * Add import of ietf-segment-routing-common module.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
```

```
     }

     revision 2016-07-07 {
       description
         "* Change routing-protocol to control-plane-protocol.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2016-03-20 {
       description
         "* Remove routing-instance.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2015-10-19 {
       description
         "* Add per-protocol SRGB support.
          * Editorial changes.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2015-09-02 {
       description
         "* Author list update.
          * Editorial changes.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     revision 2015-07-06 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
     }

     feature ti-lfa {
       description
         "Topology-Independent Loop-Free Alternate (TI-LFA)
          computation using segment routing.";
     }

     /* Groupings */
     grouping sid-sub-tlv {
       description "SID/Label sub-TLV grouping.";
```

```
       container sid-sub-tlv {
         description
           "Used to advertise the SID/Label associated with a
            prefix or adjacency.";
         leaf sid {
           type uint32;
           description
             "Segment Identifier (SID) - A 20 bit label or
              32 bit SID.";
         }
       }
     }

     grouping prefix-sid-sub-tlvs {
       description "Prefix Segment ID (SID) sub-TLVs.";
       container perfix-sid-sub-tlvs{
         description "Prefix SID sub-TLV.";
         list prefix-sid-sub-tlv {
           description "Prefix SID sub-TLV.";
           leaf flags {
             type bits {
               bit NP {
                 position 1;
                 description
                   "No-PHP flag.";
               }
               bit M {
                 position 2;
                 description
                   "Mapping server flag.";
               }
               bit E {
                 position 3;
                 description
                   "Explicit-NULL flag.";
               }
               bit V {
                 position 4;
                 description
                   "Value/Index flag.";
               }
               bit L {
                 position 5;
                 description
                   "Local flag.";
               }
             }
             description "Segment Identifier (SID) Flags.";
```

```
          }
          leaf mt-id {
            type uint8;
            description "Multi-topology ID.";
          }
          leaf algorithm {
            type uint8;
            description
              "The algorithm associated with the prefix-SID.";
          }
          leaf sid {
            type uint32;
            description "An index or label.";
          }
        }
      }
    }

    grouping extended-prefix-range-tlvs {
      description "Extended prefix range TLV grouping.";

      container extended-prefix-range-tlvs {
        description "The list of range of prefixes.";
        list extended-prefix-range-tlv {    //type=2?
          description "The range of prefixes.";
          leaf range-size {
            type uint16;
            description "The number of prefixes covered by the
                         advertisement.";
          }
          leaf flags {
            type bits {
              bit IA {
                position 0;
                description
                  "Inter-Area flag.";
              }
            }
            description "Flags.";
          }
          leaf prefix {
            type inet:ip-prefix;
            description "Address prefix.";
          }
          uses prefix-sid-sub-tlvs;
          uses ospf:unknown-tlvs;
        }
      }
```

```
     }

     grouping sr-algorithm-tlv {
       description "SR algorithm TLV grouping.";
       container sr-algorithm-tlv {
         description "All SR algorithm TLVs.";
         leaf-list sr-algorithm {
           type uint8;
           description
             "The Segment Routing (SR) algorithms that the router is
              currently using.";
         }
       }
     }

     grouping sid-range-tlvs {
       description "SID Range TLV grouping.";
       container sid-range-tlvs {
         description "List of SID range TLVs.";
         list sid-range-tlv {
           description "SID range TLV.";
           leaf range-size {
             type ospf:uint24;
             description "The SID range.";
           }
           uses sid-sub-tlv;
         }
       }
     }

     grouping local-block-tlvs {
       description "The SR local block TLV contains the
                    range of labels reserved for local SIDs.";
       container local-block-tlvs {
         description "List of SRLB TLVs.";
         list local-block-tlv {
           description "SRLB TLV.";
           leaf range-size {
             type ospf:uint24;
             description "The SID range.";
           }
           uses sid-sub-tlv;
         }
       }
     }

     grouping srms-preference-tlv {
       description "The SRMS preference TLV is used to advertise
```

```
                   a preference associated with the node that acts
                   as an SR Mapping Server.";
        container srms-preference-tlv {
          description "SRMS Preference TLV.";
          leaf preference {
            type uint8 {
              range "0 .. 255";
            }
            description "SRMS preference TLV, vlaue from 0 to 255.";
          }
        }
      }

      /* Configuration */
      augment "/rt:routing/rt:control-plane-protocols"
            + "/rt:control-plane-protocol/ospf:ospf/ospf:instance" {
        when "../../rt:type = 'ospf:ospfv2' or "
           + "../../rt:type = 'ospf:ospfv3'" {
          description
            "This augments the OSPF routing protocol when used.";
        }
        description
          "This augments the OSPF protocol configuration
           with segment routing.";
        uses sr:controlplane-cfg;
        container protocol-srgb {
          if-feature sr:protocol-srgb;
          uses sr-cmn:srgb-cfg;
          description
            "Per-protocol SRGB.";
        }
      }

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/ospf:ospf/ospf:instance/"
            + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
        when "../../../../../../rt:type = 'ospf:ospfv2' or "
           + "../../../../../../rt:type = 'ospf:ospfv3'" {
          description
            "This augments the OSPF interface configuration
             when used.";
        }
        description
          "This augments the OSPF protocol interface
           configuration with segment routing.";

        uses sr:igp-interface-cfg;
      }
```

```
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/ospf:ospf/ospf:instance/"
           + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
           + "ospf:fast-reroute" {
       when "../../../../../../../../rt:type = 'ospf:ospfv2' or "
          + "../../../../../../../../rt:type = 'ospf:ospfv3'" {
         description
           "This augments the OSPF routing protocol when used.";
       }
       description
         "This augments the OSPF protocol IP-FRR with TI-LFA.";

       container ti-lfa {
         if-feature ti-lfa;
         leaf enable {
           type boolean;
           description
            "Enables TI-LFA computation.";
         }
         description
           "Topology Independent Loop Free Alternate
            (TI-LFA) support.";
       }
     }

     /* Database */
     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:instance/ospf:areas/ospf:area/"
           + "ospf:interfaces/ospf:interface/ospf:database/"
           + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
           + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
           + "ospf:ospfv2/ospf:body/ospf:opaque/"
           + "ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv" {
       when "../../../../../../../../../../../../../../../../"
          + "rt:type = 'ospf:ospfv2'" {
         description
           "This augmentation is only valid for OSPFv2.";
       }
       description
         "SR specific TLVs for OSPFv2 extended prefix TLV
          in type 9 opaque LSA.";
       uses prefix-sid-sub-tlvs;
     }

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:instance/ospf:areas/"
```

```
            + "ospf:area/ospf:database/"
            + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
            + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
            + "ospf:ospfv2/ospf:body/ospf:opaque/"
            + "ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv" {
      when "../../../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR specific TLVs for OSPFv2 extended prefix TLV
         in type 10 opaque LSA.";
      uses prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
            + "rt:control-plane-protocols/rt:control-plane-protocol/"
            + "ospf:ospf/ospf:instance/ospf:database/"
            + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
            + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
            + "ospf:ospfv2/ospf:body/ospf:opaque/"
            + "ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv" {
      when "../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR specific TLVs for OSPFv2 extended prefix TLV
         in type 11 opaque LSA.";
      uses prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
            + "rt:control-plane-protocols/rt:control-plane-protocol/"
            + "ospf:ospf/ospf:instance/ospf:areas/"
            + "ospf:area/ospf:database/"
            + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
            + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
            + "ospf:ospfv2/ospf:body/ospf:opaque/"
            + "ospf:extended-link-tlvs/ospf:extended-link-tlv" {
      when "../../../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
```

```
          "SR specific TLVs for OSPFv2 extended link TLV
           in type 10 opaque LSA.";

        container adj-sid-sub-tlvs {
          description "Adjacency SID optional sub-TLVs.";
          list adj-sid-sub-tlv {
            description "List of Adjacency SID sub-TLVs.";
            leaf flags {
              type bits {
                bit B {
                  position 0;
                  description
                    "Backup flag.";
                }
                bit V {
                  position 1;
                  description
                    "Value/Index flag.";
                }
                bit L {
                  position 2;
                  description
                    "Local/Global flag.";
                }
                bit G {
                  position 3;
                  description
                    "Group flag.";
                }
                bit P {
                  position 4;
                  description
                    "Persistent flag.";
                }
              }
              description "Flags.";
            }
            leaf mt-id {
              type uint8;
              description "Multi-topology ID.";
            }
            leaf weight {
              type uint8;
              description "Weight used for load-balancing.";
            }
            leaf sid {
              type uint32;
              description "Segment Identifier (SID) index/label.";
```

```
          }
        }
      }

      container lan-adj-sid-sub-tlvs {
        description "LAN Adjacency SID optional sub-TLVs.";
        list lan-adj-sid-sub-tlv {
          description "List of LAN adjacency SID sub-TLVs.";
          leaf flags {
            type bits {
              bit B {
                position 0;
                description
                  "Backup flag.";
              }
              bit V {
                position 1;
                description
                  "Value/Index flag.";
              }
              bit L {
                position 2;
                description
                  "Local/Global flag.";
              }
              bit G {
                position 3;
                description
                  "Group flag.";
              }
              bit P {
                position 4;
                description
                  "Persistent flag.";
              }
            }
            description "Flags.";
          }
          leaf mt-id {
            type uint8;
            description "Multi-topology ID.";
          }
          leaf weight {
            type uint8;
            description "Weight used for load-balancing.";
          }
          leaf neighbor-router-id {
            type yang:dotted-quad;
```

```
            description "Neighbor router ID.";
          }
          leaf sid {
            type uint32;
            description "Segment Identifier (SID) index/label.";
          }
        }
      }
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:instance/ospf:areas/ospf:area/"
          + "ospf:interfaces/ospf:interface/ospf:database/"
          + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
          + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque" {
      when "../../../../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }

      description
        "SR specific TLVs for OSPFv2 type 9 opaque LSA.";

      uses extended-prefix-range-tlvs;
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:instance/ospf:areas/"
          + "ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque" {
      when "../../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }

      description
        "SR specific TLVs for OSPFv2 type 10 opaque LSA.";
```

```
       uses extended-prefix-range-tlvs;
       uses sr-algorithm-tlv;
       uses sid-range-tlvs;
       uses local-block-tlvs;
       uses srms-preference-tlv;
     }

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:instance/ospf:database/"
           + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
           + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
           + "ospf:ospfv2/ospf:body/ospf:opaque" {
       when "../../../../../../../../../../"
          + "rt:type = 'ospf:ospfv2'" {
         description
           "This augmentation is only valid for OSPFv2.";
       }
       description
         "SR specific TLVs for OSPFv2 type 11 opaque LSA.";

       uses extended-prefix-range-tlvs;
       uses sr-algorithm-tlv;
       uses sid-range-tlvs;
       uses local-block-tlvs;
       uses srms-preference-tlv;
     }
   }
   <CODE ENDS>
```

4. Security Considerations

The data model defined does not create any security implications.

This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg].

5. Acknowledgements

The authors wish to thank Yi Yang, Alexander Clemm, Gaurav Gupta, Ladislav Lhotka, Stephane Litkowski, Greg Hankins, Manish Gupta and Alan Davey for their thorough reviews and helpful comments.

This document was produced using Marshall Rose's xml2rfc tool.

6. References

6.1. Normative References

[RFC1793] Moy, J., "Extending OSPF to Support Demand Circuits", RFC 1793, DOI 10.17487/RFC1793, April 1995, <http://www.rfc-editor.org/info/rfc1793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

[RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <http://www.rfc-editor.org/info/rfc3101>.

[RFC3623] Moy, J., Pillay-Esnault, P., and A. Lindem, "Graceful OSPF Restart", RFC 3623, DOI 10.17487/RFC3623, November 2003, <http://www.rfc-editor.org/info/rfc3623>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC4577] Rosen, E., Psenak, P., and P. Pillay-Esnault, "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4577, DOI 10.17487/RFC4577, June 2006, <http://www.rfc-editor.org/info/rfc4577>.

[RFC4750] Joyal, D., Ed., Galecki, P., Ed., Giacalone, S., Ed., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, DOI 10.17487/RFC4750, December 2006, <http://www.rfc-editor.org/info/rfc4750>.

[RFC5187] Pillay-Esnault, P. and A. Lindem, "OSPFv3 Graceful Restart", RFC 5187, DOI 10.17487/RFC5187, June 2008, <http://www.rfc-editor.org/info/rfc5187>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <http://www.rfc-editor.org/info/rfc5340>.

[RFC5643] Joyal, D., Ed. and V. Manral, Ed., "Management Information Base for OSPFv3", RFC 5643, DOI 10.17487/RFC5643, August 2009, <http://www.rfc-editor.org/info/rfc5643>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <http://www.rfc-editor.org/info/rfc5838>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6565] Pillay-Esnault, P., Moyer, P., Doyle, J., Ertekin, E., and M. Lundberg, "OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol", RFC 6565, DOI 10.17487/RFC6565, June 2012, <http://www.rfc-editor.org/info/rfc6565>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <http://www.rfc-editor.org/info/rfc7950>.

[RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <http://www.rfc-editor.org/info/rfc8022>.

6.2. Informative References

[I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "Yang Data Model for OSPF Protocol", draft-ietf-ospf-yang-08 (work in progress), July 2017.

[I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-07 (work in progress), July 2017.

Appendix A. Contributors' Addreses

Dean Bogdanovic
Volta Networks, Inc.

EMail: dean@voltanet.io

Kiran Koushik Agrahara Sreenivasa
Cisco Systems
12515 Research Blvd, Bldg 4
Austin, TX 78681
USA

EMail: kkoushik@cisco.com

Authors' Addresses

Derek Yeung
Arrcus

EMail: derek@arrcus.com

Yingzhen Qu
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: yingzhen.qu@huawei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

EMail: zzhang@juniper.net

Ing-Wher Chen
Jabil Circuit

EMail: Ing-Wher_chen@jabil.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

EMail: acee@cisco.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018
D. Yeung
Arrcus
Y. Qu
Huawei
J. Zhang
Juniper Networks
I. Chen
Jabil
A. Lindem
Cisco Systems
October 30, 2017

Yang Data Model for OSPF Protocol
draft-ietf-ospf-yang-09

Abstract

This document defines a YANG data model that can be used to configure and manage OSPF.

Status of This Memo

Copyright Notice

Table of Contents

1. Overview

YANG [RFC6020] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g., ReST) and encodings other than XML (e.g., JSON) are being defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage OSPF and it is an augmentation to the core routing data model. A core routing data model is defined in [RFC8022], and it provides the basis for the development of data models for routing protocols. The interface data model is defined in [RFC7223] and is used for referencing interfaces from the routing protocol. The key-chain data model used for OSPF authentication is defined in [RFC8177] and provides both a reference to configured key-chains and an enumeration of cryptographic algorithms.

Both OSPFv2 [RFC2328] and OSPFv3 [RFC5340] are supported. In addition to the core OSPF protocol, features described in other OSPF RFCs are also supported. These includes demand circuit [RFC1793], traffic engineering [RFC3630], multiple address family [RFC5838], graceful restart [RFC3623] [RFC5187], NSSA [RFC3101], and OSPF(v3) as a PE-CE Protocol [RFC4577], [RFC6565]. These non-core features are optional in the OSPF data model.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Design of Data Model

Although the basis of OSPF configuration elements like routers, areas, and interfaces remains the same, the detailed configuration model varies among router vendors. Differences are observed in terms of how the protocol engine is tied to the routing domain, how multiple protocol engines are be instantiated among others.

The goal of this document is to define a data model that provides a common user interface to the OSPFv2 and OSPFv3 protocols. There is very little information that is designated as "mandatory", providing freedom for vendors to adapt this data model to their respective product implementations.

2.1. OSPF Operational State

The OSPF operational state is included in the same tree as OSPF configuration consistent with Network Management Datastore Architecture [I-D.ietf-netmod-revised-datastores]. Consequently, only the routing container in the ietf-routing model [RFC8022] is augmented. The routing-state container is not augmented.

2.2. Overview

The OSPF YANG module defined in this document has all the common building blocks for the OSPF protocol.

The OSPF YANG module augments the /routing/control-plane-protocols/ control-plane-protocol path defined in the ietf-routing module.

```
   module: ietf-ospf
     augment /rt:routing/rt:control-plane-protocols/
              rt:control-plane-protocol:
       +--rw ospf
             .
             .
          +--rw operation-mode?          identityref
          +--rw instance* [af]
             .
             .
             +--rw areas
             |  +--rw area* [area-id]
             |     +--rw area-id                   area-id-type
             |        .
             |        .
             |     +--rw virtual-links
             |     |  +--rw virtual-link* [transit-area-id router-id]
             |     |     .
             |     |     .
             |     +--rw sham-links {pe-ce-protocol}?
             |     |  +--rw sham-link* [local-id remote-id]
             |     |     .
             |     |     .
             |     +--rw interfaces
             |        +--rw interface* [name]
             |           .
             |           .
             +--rw topologies {multi-topology}?
                +--rw topology* [name]
                   .
                   .
```

The ospf module is intended to match to the vendor specific OSPF configuration construct that is identified by the local identifier 'name'. The field 'version' allows support for OSPFv2 and OSPFv3.

The ospf container includes one or more OSPF protocol engines, each enclosed in a separate instance entity. Each instance includes information for the routing domain based on the [routing-instance af] specification. There is no default routing domain assumed by the data model. For example, to enable OSPF on a vendor's default IPv4 routing domain, an explicit instance entity with a specification like ["default" "ipv4-unicast"] is required. The instance also contains OSPF router level configuration and operational state.

The instance/area and instance/area/interface containers respectively define the OSPF configuration and operational state for OSPF areas and interfaces.

The instance/topology container defines the OSPF configuration and operational state for OSPF topologies when the multi-topology feature is supported.

2.3. OSPFv2 and OSPFv3

The data model defined herein supports both OSPFv2 and OSPFv3.

The field 'version' is used to indicate the OSPF version and is mandatory. Based on the configured version, the data model varies to accommodate the differences between OSPFv2 and OSPFv3.

2.4. Optional Features

Optional features are beyond the basic OSPF configuration and it is the responsibility of each vendor to decide whether to support a given feature on a particular device.

This model defines a number of features, such as NSR, max-LSA, etc. It is expected that vendors will support additional features through vendor-specific augmentations.

2.5. OSPF Router Configuration/Operational State

The ospf container is the top level container in this data model. It contains shared information among the OSPF instances configured within the container.

```
module: ietf-ospf
  augment /rt:routing/rt:control-plane-protocols/
           rt:control-plane-protocol:
    +--rw ospf
       +--rw operation-mode?          identityref
       +--rw instance* [af]
          .
          .
```

2.6. OSPF Instance Configuration/Operational State

The instance container represents an OSPF protocol engine and contains the router level configuration and operational state. The routing domain for each instance is dictated through the specification of [routing-instance af]. The instance level operational state includes the instance level statistics, IETF SPF delay statistics, AS-Scoped Link State Database, local RIB, SPF Log, and the LSA log.

```
module: ietf-ospf
```

```
     augment /rt:routing/rt:control-plane-protocols/
             rt:control-plane-protocol:
       +--rw ospf
          .
          .
          +--rw instance* [af]
             +--rw af                     identityref
             +--rw explicit-router-id?   rt-types:router-id
             |                            {explicit-router-id}?
             +--rw preference
             |  +--rw (scope)?
             |     +--:(single-value)
             |     |  +--rw all?          uint8
             |     +--:(multi-values)
             |        +--rw (granularity)?
             |        |  +--:(detail)
             |        |  |  +--rw intra-area?   uint8
             |        |  |  +--rw inter-area?   uint8
             |        |  +--:(coarse)
             |        |     +--rw internal?     uint8
             |        +--rw external?     uint8
             +--rw nsr {nsr}?
             |  +--rw enable?   boolean
             +--rw graceful-restart {graceful-restart}?
             |  +--rw enable?                       boolean
             |  +--rw helper-enable?                boolean
             |  +--rw restart-interval?             uint16
             |  +--rw helper-strict-lsa-checking?   boolean
             +--rw enable?                boolean {admin-control}?
             +--rw auto-cost {auto-cost}?
             |  +--rw enable?                 boolean
             |  +--rw reference-bandwidth?   uint32
             +--rw spf-control
             |  +--rw paths?             uint16 {max-ecmp}?
             |  +--rw ietf-spf-delay {ietf-spf-delay}?
             |     +--rw initial-delay?   uint16
             |     +--rw short-delay?     uint16
             |     +--rw long-delay?      uint16
             |     +--rw hold-down?       uint16
             |     +--rw time-to-learn?   uint16
             +--rw database-control
             |  +--rw max-lsa?   uint32 {max-lsa}?
             +--rw stub-router {stub-router}?
             |  +--rw (trigger)?
             |     +--:(always)
             |        +--rw always!
             +--rw mpls
             |  +--rw te-rid {te-rid}?
```

```
             |  |  +--rw ipv4-router-id?   inet:ipv4-address
             |  |  +--rw ipv6-router-id?   inet:ipv6-address
             |  +--rw ldp
             |     +--rw igp-sync?   boolean {ldp-igp-sync}?
             +--rw fast-reroute {fast-reroute}?
             |  +--rw lfa {lfa}?
             +--rw node-tags {node-tag}?
             |  +--rw node-tag* [tag]
             |     +--rw tag    uint32
             +--ro router-id?
             +--ro local-rib
             |  +--ro route* [prefix]
             |     +--ro prefix        inet:ip-prefix
             |     +--ro next-hops
             |     |  +--ro next-hop* [next-hop]
             |     |     +--ro outgoing-interface?   if:interface-ref
             |     |     +--ro next-hop              inet:ip-address
             |     +--ro metric?       uint32
             |     +--ro route-type?   route-type
             |     +--ro route-tag?    uint32
             +--ro statistics
             |  +--ro originate-new-lsa-count?   yang:counter32
             |  +--ro rx-new-lsas-count?         yang:counter32
             |  +--ro as-scope-lsa-count?        yang:gauge32
             |  +--ro as-scope-lsa-chksum-sum?   uint32
             |  +--ro database
             |     +--ro as-scope-lsa-type*
             |        +--ro lsa-type?        uint16
             |        +--ro lsa-count?       yang:gauge32
             |        +--ro lsa-cksum-sum?   int32
             +--ro ietf-spf-delay
             |  +--ro initial-delay?             uint16
             |  +--ro short-delay?               uint16
             |  +--ro long-delay?                uint16
             |  +--ro hold-down?                 uint16
             |  +--ro time-to-learn?             uint16
             |  +--ro current-state?             enumeration
             |  +--ro remaining-time-to-learn?   uint16
             |  +--ro remaining-hold-down?       uint16
             |  +--ro last-event-received?       yang:timestamp
             |  +--ro next-spf-time?             yang:timestamp
             |  +--ro last-spf-time?             yang:timestamp
             +--ro database
             |  +--ro as-scope-lsa-type* [lsa-type]
             |     +--ro as-scope-lsas
             |        +--ro as-scope-lsa* [lsa-id adv-router]
             |           +--ro lsa-id               union
             |           +--ro adv-router           inet:ipv4-address
```

```
             |            +--ro decoded-completed?   boolean
             |            +--ro raw-data?            yang:hex-string
             |            +--ro (version)?
             |               +--:(ospfv2)
             |               |  +--ro ospfv2
             .               .
             .               .
             |               +--:(ospfv3)
             |                  +--ro ospfv3
             .
             .
             +--ro spf-log
             |  +--ro event* [id]
             |     +--ro id                     uint32
             |     +--ro spf-type?              enumeration
             |     +--ro schedule-timestamp?   yang:timestamp
             |     +--ro start-timestamp?      yang:timestamp
             |     +--ro end-timestamp?        yang:timestamp
             |     +--ro trigger-lsa*
             |        +--ro area-id?      area-id-type
             |        +--ro link-id?      union
             |        +--ro type?         uint16
             |        +--ro lsa-id?       yang:dotted-quad
             |        +--ro adv-router?   yang:dotted-quad
             |        +--ro seq-num?      uint32
             +--ro lsa-log
             |  +--ro event* [id]
             |     +--ro id                     uint32
             |     +--ro lsa
             |     |  +--ro area-id?      area-id-type
             |     |  +--ro link-id?      union
             |     |  +--ro type?         uint16
             |     |  +--ro lsa-id?       yang:dotted-quad
             |     |  +--ro adv-router?   yang:dotted-quad
             |     |  +--ro seq-num?      uint32
             |     +--ro received-timestamp?   yang:timestamp
             |     +--ro reason?               identityref
             .
             .
```

2.7. OSPF Area Configuration/Operational State

The area container contains OSPF area configuration and the list of interface containers representing all the OSPF interfaces in the area. The area operational state includes the area statistics and the area Link State Database (LSDB).

module: ietf-ospf

```
     augment /rt:routing/rt:control-plane-protocols/
              rt:control-plane-protocol:
       +--rw ospf
          .
          .
          +--rw instance* [af]
             +--rw areas
             |  +--rw area* [area-id]
             |     +--rw area-id                  area-id-type
             |     +--rw area-type?               identityref
             |     +--rw summary?                 boolean
             |     +--rw default-cost?            uint32
             |     +--rw ranges
             |     |  +--rw range* [prefix]
             |     |     +--rw prefix       inet:ip-prefix
             |     |     +--rw advertise?   boolean
             |     |     +--rw cost?        uint24
             |     +--ro statistics
             |     |  +--ro spf-runs-count?             yang:counter32
             |     |  +--ro abr-count?                  yang:gauge32
             |     |  +--ro asbr-count?                 yang:gauge32
             |     |  +--ro ar-nssa-translator-event-count?
             |     |                                    yang:counter32
             |     |  +--ro area-scope-lsa-count?       yang:gauge32
             |     |  +--ro area-scope-lsa-cksum-sum?   int32
             |     |  +--ro database
             |     |     +--ro area-scope-lsa-type*
             |     |        +--ro lsa-type?        uint16
             |     |        +--ro lsa-count?       yang:gauge32
             |     |        +--ro lsa-cksum-sum?   int32
             |     +--ro database
             |     |  +--ro area-scope-lsa-type* [lsa-type]
             |     |     +--ro lsa-type           uint16
             |     |     +--ro area-scope-lsas
             |     |        +--ro area-scope-lsa* [lsa-id adv-router]
             |     |           +--ro lsa-id                union
             .     .           .
             .     .           .
             |     |           +--ro (version)?
             |     |              +--:(ospfv2)
             |     |              |  +--ro ospfv2
             |     |              |     +--ro header
             .     .              .     .
             .     .              .     .
             |     |              |     +--ro body
             |     |              |        +--ro router
             .     .              .        .
             .     .              .        .
```

```
             |     |              |        +--ro network
             .     .              .        .
             .     .              .        .
             |     |              |        +--ro summary
             .     .              .        .
             .     .              .        .
             |     |              |        +--ro external
             .     .              .        .
             .     .              .        .
             |     |              |        +--ro opaque
             .     .              .        .
             .     .              .        .
             |     |              +--:(ospfv3)
             |     |                 +--ro ospfv3
             |     |                    +--ro header
             .     .                    .
             .     .                    .
             |     |                    +--ro body
             |     |                       +--ro router
             .     .                       .
             .     .                       .
             |     |                       +--ro network
             .     .                       .
             .     .                       .
             |     |                       +--ro inter-area-prefix
             .     .                       .
             .     .                       .
             |     |                       +--ro inter-area-router
             .     .                       .
             .     .                       .
             |     |                       +--ro as-external
             .     .                       .
             .     .                       .
             |     |                       +--ro nssa
             .     .                       .
             .     .                       .
             |     |                       +--ro link
             .     .                       .
             .     .                       .
             |     |                       +--ro intra-area-prefix
             .     .                       .
             .     .                       .
             |     |                       +--ro router-information
             .     .                       .
             .     .                       .
             |     +--rw virtual-links
             |     |  +--rw virtual-link* [transit-area-id router-id]
             |     |     +--rw transit-area-id       -> ../../../../
```

```
     |     |                                   area/area-id
     |     |     +--rw router-id             rt-types:router-id
     |     |     +--rw hello-interval?       uint16
     |     |     +--rw dead-interval?        uint32
     |     |     +--rw retransmit-interval?  uint16
     |     |     +--rw transmit-delay?       uint16
     |     |     +--rw lls?                  boolean {lls}?
     |     |     +--rw ttl-security {ttl-security}?
     |     |     |  +--rw enable?   boolean
     |     |     |  +--rw hops?     uint8
     |     |     +--rw enable?               boolean
     |     |                                 {admin-control}?
     |     |     +--rw authentication
     |     |        +--rw (auth-type-selection)?
     |     |           +--:(auth-ipsec)
     |     |               {ospfv3-authentication-ipsec}?
     |     |           |  +--rw sa?                 string
     |     |           +--:(auth-trailer-key-chain)
     |     |           |  +--rw key-chain?
     |     |                    key-chain:key-chain-ref
     |     |           +--:(auth-trailer-key)
     |     |              +--rw key?                string
     |     |              +--rw crypto-algorithm?   identityref
     |     |     +--ro cost?                 uint16
     |     |     +--ro state?                if-state-type
     |     |     +--ro hello-timer?          uint32
     |     |     +--ro wait-timer?           uint32
     |     |     +--ro dr-router-id?         rt-types:router-id
     |     |     +--ro dr-ip-addr?           inet:ip-address
     |     |     +--ro bdr-router-id?        rt-types:router-id
     |     |     +--ro bdr-ip-addr?          inet:ip-address
     |     |     +--ro statistics
     |     |     |  +--ro if-event-count?        yang:counter32
     |     |     |  +--ro link-scope-lsa-count?  yang:gauge32
     |     |     |  +--ro link-scope-lsa-cksum-sum?
     |     |     |                               uint32
     |     |     |  +--ro database
     |     |     |     +--ro link-scope-lsa-type*
     |     |     |        +--ro lsa-type?        uint16
     |     |     |        +--ro lsa-count?       yang:gauge32
     |     |     |        +--ro lsa-cksum-sum?   int32
     |     |     +--ro neighbors
     |     |     |  +--ro neighbor* [neighbor-router-id]
     |     |     |     +--ro neighbor-router-id
     |     |     |                           rt-types:router-id
     |     |     |     +--ro address?        inet:ip-address
     |     |     |     +--ro dr-router-id?   rt-types:router-id
     |     |     |     +--ro dr-ip-addr?     inet:ip-address
```

```
             |     |     |     +--ro bdr-router-id?  rt-types:router-id
             |     |     |     +--ro bdr-ip-addr?    inet:ip-address
             |     |     |     +--ro state?          nbr-state-type
             |     |     |     +--ro dead-timer?     uint32
             |     |     |     +--ro statistics
             |     |     |        +--ro nbr-event-count?
             |     |     |                          yang:counter32
             |     |     |        +--ro nbr-retrans-qlen?
             |     |     |                          yang:gauge32
             |     |     +--ro database
             |     |        +--ro link-scope-lsa-type* [lsa-type]
             |     |           +--ro lsa-type           uint16
             |     |           +--ro link-scope-lsas
             .     .
             .     .
             |     +--rw sham-links {pe-ce-protocol}?
             |     |  +--rw sham-link* [local-id remote-id]
             |     |     +--rw local-id                inet:ip-address
             |     |     +--rw remote-id               inet:ip-address
             |     |     +--rw hello-interval?         uint16
             |     |     +--rw dead-interval?          uint32
             |     |     +--rw retransmit-interval?    uint16
             |     |     +--rw transmit-delay?         uint16
             |     |     +--rw lls?                    boolean {lls}?
             |     |     +--rw ttl-security {ttl-security}?
             |     |     |  +--rw enable?   boolean
             |     |     |  +--rw hops?     uint8
             |     |     +--rw enable?              boolean
             |     |                                {admin-control}?
             |     |     +--rw authentication
             |     |     |  +--rw (auth-type-selection)?
             |     |     |     +--:(auth-ipsec)
             |     |     |         {ospfv3-authentication-ipsec}?
             |     |     |     |  +--rw sa?                  string
             |     |     |     +--:(auth-trailer-key-chain)
             |     |     |     |  +--rw key-chain?
             |     |     |              key-chain:key-chain-ref
             |     |     |     +--:(auth-trailer-key)
             |     |     |        +--rw key?                 string
             |     |     |        +--rw crypto-algorithm?    identityref
             |     |     +--rw cost?                uint16
             |     |     +--rw mtu-ignore?          boolean
             |     |                                {mtu-ignore}?
             |     |     +--rw prefix-suppression? boolean
             |     |                                {prefix-suppression}?
             |     |     +--ro state?               if-state-type
             |     |     +--ro hello-timer?         uint32
             |     |     +--ro wait-timer?          uint32
```

```
             |     |     +--ro dr-router-id?       rt-types:router-id
             |     |     +--ro dr-ip-addr?         inet:ip-address
             |     |     +--ro bdr-router-id?      rt-types:router-id
             |     |     +--ro bdr-ip-addr?        inet:ip-address
             |     |     +--ro statistics
             |     |     |  +--ro if-event-count?        yang:counter32
             |     |     |  +--ro link-scope-lsa-count?  yang:gauge32
             |     |     |  +--ro link-scope-lsa-cksum-sum?
             |     |     |                               uint32
             |     |     |  +--ro database
             |     |     |     +--ro link-scope-lsa-type*
             |     |     |        +--ro lsa-type?        uint16
             |     |     |        +--ro lsa-count?       yang:gauge32
             |     |     |        +--ro lsa-cksum-sum?   int32
             |     |     +--ro neighbors
             |     |     |  +--ro neighbor* [neighbor-router-id]
             |     |     |     +--ro neighbor-router-id
             |     |     |                             rt-types:router-id
             |     |     |     +--ro address?          inet:ip-address
             |     |     |     +--ro dr-router-id?     rt-types:router-id
             |     |     |     +--ro dr-ip-addr?       inet:ip-address
             |     |     |     +--ro bdr-router-id?    rt-types:router-id
             |     |     |     +--ro bdr-ip-addr?      inet:ip-address
             |     |     |     +--ro state?            nbr-state-type
             |     |     |     +--ro dead-timer?       uint32
             |     |     |     +--ro statistics
             |     |     |        +--ro nbr-event-count?
             |     |     |                             yang:counter32
             |     |     |        +--ro nbr-retrans-qlen?
             |     |     |                             yang:gauge32
             |     |     +--ro database
             |     |        +--ro link-scope-lsa-type* [lsa-type]
             |     |           +--ro lsa-type           uint16
             |     |           +--ro link-scope-lsas
             .     .
             .     .
```

2.8. OSPF Interface Configuration/Operational State

The interface container contains OSPF interface configuration and operational state. The interface operational state includes the statistics, list of neighbors, and link-local Link State database (LSDB).

```
module: ietf-ospf
  augment /rt:routing/rt:control-plane-protocols/
           rt:control-plane-protocol:
    +--rw ospf
```

```
          .
          .
          +--rw instance* [af]
             .
             .
             +--rw areas
             |  +--rw area* [area-id]
             |     .
             |     .
             |     +--rw interfaces
             |        +--rw interface* [name]
             |           +--rw name                   if:interface-ref
             |           +--rw interface-type?        enumeration
             |           +--rw passive?               boolean
             |           +--rw demand-circuit?        boolean
             |                                        {demand-circuit}?
             |           +--rw priority?              uint8
             |           +--rw multi-areas {multi-area-adj}?
             |           |  +--rw multi-area* [multi-area-id]
             |           |     +--rw multi-area-id    area-id-type
             |           |     +--rw cost?            uint16
             |           +--rw static-neighbors
             |           |  +--rw neighbor* [identifier]
             |           |     +--rw identifier       inet:ip-address
             |           |     +--rw cost?            uint16
             |           |     +--rw poll-interval?   uint16
             |           |     +--rw priority?        uint8
             |           +--rw node-flag?             boolean
             |                                        {node-flag}?
             |           +--rw bfd {bfd}?
             |           |  +--rw enable?   boolean
             |           +--rw fast-reroute {fast-reroute}?
             |           |  +--rw lfa {lfa}?
             |           |     +--rw candidate-enable?   boolean
             |           |     +--rw enable?             boolean
             |           |     +--rw remote-lfa {remote-lfa}?
             |           |        +--rw enable?   boolean
             |           +--rw hello-interval?        uint16
             |           +--rw dead-interval?         uint32
             |           +--rw retransmit-interval?   uint16
             |           +--rw transmit-delay?        uint16
             |           +--rw lls?                   boolean {lls}?
             |           +--rw ttl-security {ttl-security}?
             |           |  +--rw enable?   boolean
             |           |  +--rw hops?     uint8
             |           +--rw enable?                boolean
             |                                        {admin-control}?
             |           +--rw authentication
```

```
            |           |  +--rw (auth-type-selection)?
            |           |     +--:(auth-ipsec)
            |           |     |   {ospfv3-authentication-ipsec}?
            |           |     |  +--rw sa?                   string
            |           |     +--:(auth-trailer-key-chain)
            |           |     |  +--rw key-chain?
            |           |     |        key-chain:key-chain-ref
            |           |     +--:(auth-trailer-key)
            |           |        +--rw key?                  string
            |           |        +--rw crypto-algorithm?   identityref
            |           +--rw cost?                 uint16
            |           +--rw mtu-ignore?           boolean
            |           |                           {mtu-ignore}?
            |           +--rw prefix-suppression? boolean
            |           |                           {prefix-suppression}?
            |           +--ro state?                   if-state-type
            |           +--ro hello-timer?          uint32
            |           +--ro wait-timer?           uint32
            |           +--ro dr-router-id?         rt-types:router-id
            |           +--ro dr-ip-addr?           inet:ip-address
            |           +--ro bdr-router-id?        rt-types:router-id
            |           +--ro bdr-ip-addr?          inet:ip-address
            |           +--ro statistics
            |           |  +--ro if-event-count?        yang:counter32
            |           |  +--ro link-scope-lsa-count?  yang:gauge32
            |           |  +--ro link-scope-lsa-cksum-sum?
            |           |                               uint32
            |           |  +--ro database
            |           |     +--ro link-scope-lsa-type*
            |           |        +--ro lsa-type?        uint16
            |           |        +--ro lsa-count?       yang:gauge32
            |           |        +--ro lsa-cksum-sum?   int32
            |           +--ro neighbors
            |           |  +--ro neighbor* [neighbor-router-id]
            |           |     +--ro neighbor-router-id
            |           |                             rt-types:router-id
            |           |     +--ro address?        inet:ip-address
            |           |     +--ro dr-router-id?   rt-types:router-id
            |           |     +--ro dr-ip-addr?     inet:ip-address
            |           |     +--ro bdr-router-id?  rt-types:router-id
            |           |     +--ro bdr-ip-addr?    inet:ip-address
            |           |     +--ro state?          nbr-state-type
            |           |     +--ro dead-timer?     uint32
            |           |     +--ro statistics
            |           |        +--ro nbr-event-count?
            |           |                           yang:counter32
            |           |        +--ro nbr-retrans-qlen?
            |           |                           yang:gauge32
```

```
             |           +--ro database
             |           .  +--ro link-scope-lsa-type* [lsa-type]
             |           .     +--ro lsa-type           uint16
             |           .     +--ro link-scope-lsas
             .           .
             .           .
             |           +--rw topologies {ospf:multi-topology}?
             |           |  +--rw topology* [name]
             |           |     +--rw name  -> ../../../../../../../../
             |           |                    ../../../rt:ribs/rib/name
             |           |     +--rw cost? uint32
             |           +--rw instance-id?           uint8
             .
             .
```

2.9. OSPF notification

This YANG model defines a list of notifications that inform YANG clients of important events detected during protocol operation. The defined notifications cover the common set of traps from the OSPFv2 MIB [RFC4750] and OSPFv3 MIB [RFC5643].

```
     notifications:
       +---n if-state-change
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro (if-link-type-selection)?
       |  |  +--:(interface)
       |  |  |  +--ro interface
       |  |  |     +--ro interface?   if:interface-ref
       |  |  +--:(virtual-link)
       |  |  |  +--ro virtual-link
       |  |  |     +--ro transit-area-id?      area-id-type
       |  |  |     +--ro neighbor-router-id?   rt-types:router-id
       |  |  +--:(sham-link)
       |  |     +--ro sham-link
       |  |        +--ro area-id?          area-id-type
       |  |        +--ro local-ip-addr?    inet:ip-address
       |  |        +--ro remote-ip-addr?   inet:ip-address
       |  +--ro state?                  if-state-type
       +---n if-config-error
       |  +--ro routing-protocol-name?
```

```
        |  +     -> /rt:routing/control-plane-protocols/
        |  +         control-plane-protocol/name
        |  +--ro af?
        |  +     -> /rt:routing/control-plane-protocols/
        |  +         control-plane-protocol
        |  +         [rt:name=current()/../routing-protocol-name]/
        |  +         ospf:ospf/instance/af
        |  +--ro (if-link-type-selection)?
        |  |  +--:(interface)
        |  |  |  +--ro interface
        |  |  |     +--ro interface?   if:interface-ref
        |  |  +--:(virtual-link)
        |  |  |  +--ro virtual-link
        |  |  |     +--ro transit-area-id?      area-id-type
        |  |  |     +--ro neighbor-router-id?   rt-types:router-id
        |  |  +--:(sham-link)
        |  |     +--ro sham-link
        |  |        +--ro area-id?          area-id-type
        |  |        +--ro local-ip-addr?    inet:ip-address
        |  |        +--ro remote-ip-addr?   inet:ip-address
        |  +--ro packet-source?           yang:dotted-quad
        |  +--ro packet-type?             packet-type
        |  +--ro error?                   enumeration
        +---n nbr-state-change
        |  +--ro routing-protocol-name?
        |  +     -> /rt:routing/control-plane-protocols/
        |  +         control-plane-protocol/name
        |  +--ro af?
        |  +     -> /rt:routing/control-plane-protocols/
        |  +         control-plane-protocol
        |  +         [rt:name=current()/../routing-protocol-name]/
        |  +         ospf:ospf/instance/af
        |  +--ro (if-link-type-selection)?
        |  |  +--:(interface)
        |  |  |  +--ro interface
        |  |  |     +--ro interface?   if:interface-ref
        |  |  +--:(virtual-link)
        |  |  |  +--ro virtual-link
        |  |  |     +--ro transit-area-id?      area-id-type
        |  |  |     +--ro neighbor-router-id?   rt-types:router-id
        |  |  +--:(sham-link)
        |  |     +--ro sham-link
        |  |        +--ro area-id?          area-id-type
        |  |        +--ro local-ip-addr?    inet:ip-address
        |  |        +--ro remote-ip-addr?   inet:ip-address
        |  +--ro neighbor-router-id?      rt-types:router-id
        |  +--ro neighbor-ip-addr?        yang:dotted-quad
        |  +--ro state?                   nbr-state-type
```

```
       +---n nbr-restart-helper-status-change
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro (if-link-type-selection)?
       |  |  +--:(interface)
       |  |  |  +--ro interface
       |  |  |     +--ro interface?   if:interface-ref
       |  |  +--:(virtual-link)
       |  |  |  +--ro virtual-link
       |  |  |     +--ro transit-area-id?      area-id-type
       |  |  |     +--ro neighbor-router-id?   rt-types:router-id
       |  |  +--:(sham-link)
       |  |     +--ro sham-link
       |  |        +--ro area-id?          area-id-type
       |  |        +--ro local-ip-addr?    inet:ip-address
       |  |        +--ro remote-ip-addr?   inet:ip-address
       |  +--ro neighbor-router-id?      rt-types:router-id
       |  +--ro neighbor-ip-addr?        yang:dotted-quad
       |  +--ro status?                  restart-helper-status-type
       |  +--ro age?                     uint32
       |  +--ro exit-reason?             restart-exit-reason-type
       +---n if-rx-bad-packet
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro (if-link-type-selection)?
       |  |  +--:(interface)
       |  |  |  +--ro interface
       |  |  |     +--ro interface?   if:interface-ref
       |  |  +--:(virtual-link)
       |  |  |  +--ro virtual-link
       |  |  |     +--ro transit-area-id?      area-id-type
       |  |  |     +--ro neighbor-router-id?   rt-types:router-id
       |  |  +--:(sham-link)
       |  |     +--ro sham-link
       |  |        +--ro area-id?          area-id-type
       |  |        +--ro local-ip-addr?    inet:ip-address
```

```
       |  |        +--ro remote-ip-addr?   inet:ip-address
       |  +--ro packet-source?          yang:dotted-quad
       |  +--ro packet-type?            packet-type
       +---n lsdb-approaching-overflow
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro ext-lsdb-limit?         uint32
       +---n lsdb-overflow
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro ext-lsdb-limit?         uint32
       +---n nssa-translator-status-change
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/instance/af
       |  +--ro area-id?                area-id-type
       |  +--ro status?                 nssa-translator-state-type
       +---n restart-status-change
          +--ro routing-protocol-name?
          +     -> /rt:routing/control-plane-protocols/
          +         control-plane-protocol/name
          +--ro af?
          +     -> /rt:routing/control-plane-protocols/
          +         control-plane-protocol
          +         [rt:name=current()/../routing-protocol-name]/
          +         ospf:ospf/instance/af
          +--ro status?                 restart-status-type
          +--ro restart-interval?       uint16
          +--ro exit-reason?            restart-exit-reason-type
```

2.10. OSPF RPC Operations

The "ietf-ospf" module defines two RPC operations:

- o clear-database: reset the content of a particular OSPF database.

- o clear-neighbor: restart a particular set of OSPF neighbor.

```
  rpcs:
    +---x clear-neighbor
    |  +---w input
    |     +---w routing-protocol-name
    |     +     -> /rt:routing/control-plane-protocols/
    |     +         control-plane-protocol/name
    |     +---w interface?               if:interface-ref
    +---x clear-database
       +---w input
          +---w routing-protocol-name
                -> /rt:routing/control-plane-protocols/
                    control-plane-protocol/name
```

3. OSPF Yang Module

```
<CODE BEGINS> file "ietf-ospf@2017-10-30.yang"
module ietf-ospf {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf";

  prefix ospf;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

  import iana-routing-types {
    prefix "iana-rt-types";
  }
```

```
     import ietf-routing {
       prefix "rt";
     }

     import ietf-key-chain {
       prefix "key-chain";
     }

     organization
         "IETF OSPF - OSPF Working Group";

     contact
       "WG Web:   <http://datatracker.ietf.org/group/ospf/>
        WG List:  <mailto:ospf@ietf.org>

        Editor:   Derek Yeung
                  <mailto:derek@arrcus.com>
        Author:   Acee Lindem
                  <mailto:acee@cisco.com>
        Author:   Yingzhen Qu
                  <mailto:yingzhen.qu@huawei.com>
        Author:   Jeffrey Zhang
                  <mailto:zzhang@juniper.net>
        Author:   Ing-Wher Chen
                  <mailto:ing-wher_chen@jabil.com>
        Author:   Dean Bogdanovic
                  <mailto:ivandean@gmail.com>
        Author:   Kiran Agrahara Sreenivasa
                  <mailto:kkoushik@cisco.com>";

     description
        "This YANG module defines the generic configuration and
         operational state for the OSPF protocol common to all
         vendor implementations. It is intended that the module
         will be extended by vendors to define vendor-specific
         OSPF configuration parameters and policies,
         for example route maps or route policies.

         Copyright (c) 2017 IETF Trust and the persons identified as
         authors of the code.  All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (http://trustee.ietf.org/license-info).
```

```
          This version of this YANG module is part of RFC XXXX;
          see the RFC itself for full legal notices.";

     reference "RFC XXXX";

     revision 2017-10-30 {
       description
         "* Address iana-routing-types address-family type change.
          * Complete NMDA change for routing-state augmentation.
         ";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2017-07-01 {
       description
         "* Restructure model to conform to NMDA.
          * Remove features for instance, area and interface
            inheritance.
          * Update static neighbor identifier description to
            allow for router-id, ipv4-address, and ipv6-address.
          * Added spf-log and lsa-log.
          * Use dotted-quad for OSPFv2 LSA ID.
          * Fix virtual-link transit-area-id leafref path and
            must statement.
         ";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2017-03-12 {
       description
         "* Update authors information.
          * Rename admin distance to preference.
          * Rename network type to interface type.
          * Add ietf-spf-delay as a feature.
          * Add node-tag as a feature and update LSA
            definition accordingly.
          * Remove LDP IGP autoconfig.
          * Add BFD as a feature instead of a separate module.
          * Change TE router ID to support IPv4 and IPv6 router ID.
          * Replace key-chain:crypto-algorithm-types with
            key-chain:crypto-algorithm.
          * Remove type ieee-bandwidth.
          * Import ietf-routing-types and make use of
            router-id, address-family and bandwidth-ieee-float32
            type definitions.
          * Simplify notification header.
```

```
          * Fix compilation issue in multiple must statements.
         ";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2016-10-31 {
       description
         "* Update authors information.
          * Rename candidate-disabled to candidiate-enable
            and set the default value to TRUE.
          * Rename node identifiers that end with
            'enabled' to 'enable'.
          * Set the default value of
            ospf/instance/areas/area/interfaces/interface/
            fast-reroute/lfa/enable (previously named 'enabled')
            to FALSE.
          * Set the default value of
            ospf/instance/areas/area/interfaces/interface/
            fast-reroute/remote-lfa/enable (previously named 'enabled')
            to FALSE.
          * Rename
            ospf/instance/areas/area/interfaces/interface/
            static-neighbors/neighbor/address to 'identifier'
            with type inet:ip-address
          * Add 'dead-timer' to
            ospf-state/instance/areas/area/interfaces/interface/
            neighbors/neighbor.
          * Remove 'mtu-ignore' and 'prefix-suppression' from
            virtual-link configuration.
          * Adjust range specifications from 'transmit-delay',
            'dead-interval', and 'retransmit-interval' in
            ospf/instance/areas/area/interfaces/interface.
          * Change the type of
            ospf/instance/areas/area/interface/interface/dead-interval
            to uint32 to match RFC2328 Appendix A.3.2.
          * Change hello-timer and wait-timer unit to seconds.
          * Update hello-timer, dead-timer and wait-timer descriptions.
          * Add IEEE bandwidth type and update all TE bandwidth fields
            to use it.
          * Add Nt-bit to OSPFv2 router LSA.
          * Remove L-bit from OSPFv2 router LSA.
         ";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2016-07-07 {
```

```
    description
      "* Add ospfv3 AF bit.
       * Add ospfv2 MT, L, O, DN bit.
       * Add interface priority config.
       * Change bdr-ip-address to type ip-address.
       * Rename leaf interface to name.
       * Rename rx-bad-packet to if-rx-bad-packet.
       * Move virtual link placement to backbone area.
       * Remove cost configuration from virtual link.
       * Move if-feature multi-area-adj statement.
       * Add type checksum16-type.
       * Change LSA header checksum to use checksum16-type.
       * Change routing-protocol to control-plane-protocol.
       * Change import module name to ietf-key-chain.";
    reference
      "RFC XXXX: A YANG Data Model for OSPF.";
  }

  revision 2016-03-20 {
    description
      "* Reorganize *-config and *-operation groupings.
       * Use *-config under state tree for applied config.
       * Rename config router-id to explicit-router-id.
       * Rename feature router-id to explicit-router-id.
       * Add OSPFv3 instance ID.
       * Add OSPFv3 interface ID.
       * Add ip-address for DR and BDR.
       * Remove routing-instance.
       * Change import module name to ietf-routing-key-chain.";
    reference
      "RFC XXXX: A YANG Data Model for OSPF.";
  }

  revision 2015-10-19 {
    description
      "* Remove the abstract identity ospf.
       * Make area-id-type dotted-quad only.
       * Use area-id-type for all area-id leafs.
       * Restructure notifications.
       * Move BFD support to the new ietf-ospf-bfd module.
       * Update author information.
       * Editorial changes.";
    reference
      "RFC XXXX: A YANG Data Model for OSPF.";
  }

  revision 2015-09-02 {
    description
```

```
         "* Author information update.
          * Editorial changes";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2015-07-06 {
       description
         "* Remove support for protocol-centric config.
          * Enclose list in container, except for instance.
          * Replace protocol-shutdown with admin-control.
          * Add IP-FRR per-interface config.
          * Reorganize max-path etc node.
          * Add node-flag.
          * Align config/operation hierarchy.
          * Use relative path for reference to rib.
          * Add ability to set single admin distance.
          * Make unreserved bandwidth into list.
          * Add F and T bit to OSPFv3 external LSA.
          * Remove key statement inside LSA body.
          * Add stub router support.
          * Fix usage of af-area-config.
          * Add statistics to operation data.
          * Add local rib.
          * Use dotted-quad for all router-id fields.
          * Support more than one multi-area per interface.
          * Use uint16 for LSA type.
          * Update grouping notification-instance-hdr.
          * Rework condition for opaque type and id in OSPFv2 LSA.
          * Rename local-remote-ipv4-addr with remote-if-ipv4-addr.
          * Add virtual-link/sham-link to operation state.
          * Allow multiple link TLVs in one LSA.
          * Fix bug in as-scope-lsas.
          * Remove OSPFv3 restriction in link-scope-lsas.
          * Editorial changes.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     revision 2015-03-09 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for OSPF.";
     }

     feature multi-topology {
       description
```

```
        "Support Multiple-Topolgy Routing (MTR).";
    }

    feature multi-area-adj {
      description
        "OSPF multi-area adjacency support as in RFC 5185.";
    }
    feature explicit-router-id {
      description
        "Set Router-ID per instance explicitly.";
    }

    feature demand-circuit {
      description
        "OSPF demand circuit support as in RFC 1793.";
    }

    feature mtu-ignore {
      description
        "Disable OSPF Database Description packet MTU
         mismatch checking.";
    }

    feature lls {
      description
        "OSPF link-local signaling (LLS) as in RFC 5613.";
    }

    feature prefix-suppression {
      description
        "OSPF prefix suppression support as in RFC 6860.";
    }

    feature ttl-security {
      description
        "OSPF TTL security check.";
    }

    feature nsr {
      description
        "Non-Stop-Routing (NSR).";
    }

    feature graceful-restart {
      description
        "Graceful OSPF Restart as defined in RFC 3623 and
         RFC 5187.";
    }
```

```
     feature admin-control {
       description
         "Administrative control of the protocol state.";
     }

     feature auto-cost {
       description
         "Calculate OSPF interface cost according to
          reference bandwidth.";
     }

     feature max-ecmp {
       description
         "Setting maximum number of ECMP paths.";
     }

     feature max-lsa {
       description
         "Setting maximum number of LSAs the OSPF instance
          will accept.";
     }

     feature te-rid {
       description
         "TE Router-ID.";
     }

     feature ldp-igp-sync {
       description
         "LDP IGP synchronization.";
     }

     feature ospfv3-authentication-ipsec {
       description
         "Use IPsec for OSPFv3 authentication.";
     }

     feature fast-reroute {
      description
        "Support of IP Fast Reroute (IP-FRR).";
     }

     feature node-flag {
      description
        "Support of node flag.";
     }

     feature node-tag {
```

```
      description
        "Support of node tag.";
    }

    feature lfa {
     description
       "Support of Loop Free Alternates (LFAs).";
    }

    feature remote-lfa {
     description
       "Support of Remote Loop Free Alternates (R-LFA).";
    }

    feature stub-router {
      description
        "Support of RFC 6987 OSPF Stub Router Advertisement.";
    }

    feature pe-ce-protocol {
      description
        "Support PE-CE protocol";
    }

    feature ietf-spf-delay {
      description
        "Support of IETF SPF delay algorithm.";
    }

    feature bfd {
      description
        "Support of BFD.";
    }

    identity ospfv2 {
      base "rt:routing-protocol";
      description "OSPFv2";
    }

    identity ospfv3 {
      base "rt:routing-protocol";
      description "OSPFv3";
    }

    identity operation-mode {
      description
        "OSPF operation mode.";
    }
```

```
     identity ships-in-the-night {
       base operation-mode;
       description
         "Ships-in-the-night operation mode in which
         each OSPF instance carries only one address family";
     }

     identity area-type {
       description "Base identity for OSPF area type.";
     }

     identity normal {
       base area-type;
       description "OSPF normal area.";
     }

     identity stub {
       base area-type;
       description "OSPF stub area.";
     }

     identity nssa {
       base area-type;
       description "OSPF NSSA area.";
     }

     identity lsa-log-reason {
       description
         "Base identity for an LSA log reason.";
     }

     identity lsa-refresh {
       base lsa-log-reason;
       description
         "Identity used when the LSA is logged
          as a result of receiving a refresh LSA.";
     }

     identity lsa-content-change {
       base lsa-log-reason;
       description
         "Identity used when the LSA is logged
          as a result of a change in the content
          of the LSA.";
     }

     identity lsa-purge {
       base lsa-log-reason;
```

```
       description
         "Identity used when the LSA is logged
          as a result of being purged.";
     }

     typedef uint24 {
       type uint32 {
         range "0 .. 16777215";
       }
       description
         "24-bit unsigned integer.";
     }

     typedef area-id-type {
       type yang:dotted-quad;
       description
         "Area ID type.";
     }

     typedef route-type {
       type enumeration {
         enum intra-area {
           description "OSPF intra-area route.";
         }
         enum inter-area {
           description "OSPF inter-area route.";
         }
         enum external-1 {
           description "OSPF type 1 external route.";
         }
         enum external-2 {
           description "OSPF type 2 external route.";
         }
         enum nssa-1 {
           description "OSPF type 1 NSSA route.";
         }
         enum nssa-2 {
           description "OSPF type 2 NSSA route.";
         }
       }
       description "OSPF route type.";
     }

     typedef if-state-type {
       type enumeration {
         enum Down {
           value "1";
           description
```

```
            "Interface down state.";
        }
        enum Loopback {
          value "2";
          description
            "Interface loopback state.";
        }
        enum Waiting {
          value "3";
          description
            "Interface waiting state.";
        }
        enum Point-to-Point {
          value "4";
          description
            "Interface point-to-point state.";
        }
        enum DR {
          value "5";
          description
            "Interface Designated Router (DR) state.";
        }
        enum BDR {
          value "6";
          description
            "Interface Backup Designated Router (BDR) state.";
        }
        enum DR-Other {
          value "7";
          description
            "Interface Other Designated Router state.";
        }
      }
      description
        "OSPF interface state type.";
    }

    typedef nbr-state-type {
      type enumeration {
        enum Down {
          value "1";
          description
            "Neighbor down state.";
        }
        enum Attempt {
          value "2";
          description
            "Neighbor attempt state.";
```

```
         }
         enum Init {
           value "3";
           description
             "Neighbor init state.";
         }
         enum 2-Way {
           value "4";
           description
             "Neighbor 2-Way state.";
         }
         enum ExStart {
           value "5";
           description
             "Neighbor exchange start state.";
         }
         enum Exchange {
           value "6";
           description
             "Neighbor exchange state.";
         }
         enum Loading {
           value "7";
           description
             "Neighbor loading state.";
         }
         enum Full {
           value "8";
           description
             "Neighbor full state.";
         }
       }
       description
         "OSPF neighbor state type.";
     }

     typedef restart-helper-status-type {
       type enumeration {
         enum Not-Helping {
           value "1";
           description
             "Restart helper status not helping.";
         }
         enum Helping {
           value "2";
           description
             "Restart helper status helping.";
         }
```

```
      }
      description
        "Restart helper status type.";
    }

    typedef restart-exit-reason-type {
      type enumeration {
        enum None {
          value "1";
          description
            "Not attempted.";
        }
        enum InProgress {
          value "2";
          description
            "Restart in progress.";
        }
        enum Completed {
          value "3";
          description
            "Successfully completed.";
        }
        enum TimedOut {
          value "4";
          description
            "Timed out.";
        }
        enum TopologyChanged {
          value "5";
          description
            "Aborted due to topology change.";
        }
      }
      description
        "Describes the outcome of the last attempt at a
        graceful restart, either by itself or acting
        as a helper.";
    }

    typedef packet-type {
      type enumeration {
        enum Hello {
          value "1";
          description
            "OSPF hello packet.";
        }
        enum Database-Descripton {
          value "2";
```

```
          description
            "OSPF database description packet.";
        }
        enum Link-State-Request {
          value "3";
          description
            "OSPF link state request packet.";
        }
        enum Link-State-Update {
          value "4";
          description
            "OSPF link state update packet.";
        }
        enum Link-State-Ack {
          value "5";
          description
            "OSPF link state acknowlegement packet.";
        }
      }
      description
        "OSPF packet type.";
    }

    typedef nssa-translator-state-type {
      type enumeration {
        enum Enabled {
          value "1";
          description
            "NSSA translator enabled state.";
        }
        enum Elected {
          description
            "NSSA translator elected state.";
        }
        enum Disabled {
          value "3";
          description
            "NSSA translator disabled state.";
        }
      }
      description
        "OSPF NSSA translator state type.";
    }

    typedef restart-status-type {
      type enumeration {
        enum Not-Restarting {
          value "1";
```

```
           description
             "Router is not restarting.";
         }
         enum Planned-Restart {
           description
             "Router is going through planned restart.";
         }
         enum Unplanned-Restart {
           value "3";
           description
             "Router is going through unplanned restart.";
         }
       }
       description
         "OSPF graceful restart status type.";
     }

     typedef checksum16-type {
       type string {
          pattern '(0x)?[0-9a-fA-F]{4}';
       }
       description
         "16-bit checksum in hex-string format 0xXXXX.";
     }

     grouping tlv {
       description
         "TLV";
       leaf type {
         type uint16;
         description "TLV type.";
       }
       leaf length {
         type uint16;
         description "TLV length.";
       }
       leaf value {
         type yang:hex-string;
         description "TLV value.";
       }
     }

     grouping unknown-tlvs {
       description
         "Unknown TLVs grouping - Used for unknown TLVs or
          unknown sub-TLVs.";
       container unknown-tlvs {
         description "All unknown TLVs.";
```

```
         list unknown-tlv {
           description "Unknown TLV.";
           uses tlv;
         }
       }
     }

     grouping node-tag-tlv {
       description "OSPF Node Admin Tag TLV grouping.";
       list node-tag {
         leaf tag {
           type uint32;
           description
             "Node tag value.";
         }
         description
           "List of tags.";
       }
     }

     grouping ospfv2-router-link {
       description "OSPFv2 router link.";
       leaf link-id {
         type union {
           type inet:ipv4-address;
           type yang:dotted-quad;
         }
         description "Link ID.";
       }
       leaf link-data {
         type union {
           type inet:ipv4-address;
           type uint32;
         }
         description "Link data.";
       }
       leaf type {
         type uint8;
         description "Link type.";
       }
     }

     grouping ospfv2-lsa-body {
       description "OSPFv2 LSA body.";
       container router {
         when "../../header/type = 1" {
           description
             "Only applies to Router LSAs.";
```

```
         }
         description
           "Router LSA.";
         leaf flags {
           type bits {
             bit V {
               description
                 "When set, the router is an endpoint of one or
                  more virtual links.";
             }
             bit E {
               description
                 "When set, the router is an AS Boundary Router
                  (ASBR).";
             }
             bit B {
               description
                 "When set, the router is an Area Border
                  Router (ABR).";
             }
             bit Nt {
               description
                 "When set, the router is an NSSA border router
                  that is unconditionally translating NSSA LSAs
                  into AS-external LSAs.";
             }
           }
           description "Flags.";
         }
         leaf num-of-links {
           type uint16;
           description "Number of links.";
         }
         container links {
           description "All router Links.";
           list link {
             description "Router LSA link.";
             uses ospfv2-router-link;
             container topologies {
               description "All topologies for the link.";
               list topology {
                 description
                   "Topology specific information.";
                 leaf mt-id {
                   type uint8;
                   description
                     "The MT-ID for the topology enabled on
                      the link.";
```

```
                 }
                 leaf metric {
                   type uint16;
                   description "Metric for the topology.";
                 }
               }
             }
           }
         }
       }
       container network {
         when "../../header/type = 2" {
           description
             "Only applies to Network LSAs.";
         }
         description
           "Network LSA.";
         leaf network-mask {
           type inet:ipv4-address;
           description
             "The IP address mask for the network.";
         }
         container attached-routers {
           description "All attached routers.";
           leaf-list attached-router {
             type yang:dotted-quad;
             description
               "List of the routers attached to the network.";
           }
         }
       }
       container summary {
         when "../../header/type = 3 or "
            + "../../header/type = 4" {
           description
             "Only applies to Summary LSAs.";
         }
         description
           "Summary LSA.";
         leaf network-mask {
           type inet:ipv4-address;
           description
             "The IP address mask for the network";
         }
         container topologies {
           description "All topologies for the summary LSA.";
           list topology {
             description
```

```
                "Topology specific information.";
              leaf mt-id {
                type uint8;
                description
                  "The MT-ID for the topology enabled for
                   the summary.";
              }
              leaf metric {
                type uint24;
                description "Metric for the topology.";
              }
            }
          }
        }
        container external {
          when "../../header/type = 5 or "
             + "../../header/type = 7" {
            description
              "Only applies to AS-external LSAs and NSSA LSAs.";
          }
          description
            "External LSA.";
          leaf network-mask {
            type inet:ipv4-address;
            description
              "The IP address mask for the network";
          }
          container topologies {
            description "All topologies for the external.";
            list topology {
              description
                "Topology specific information.";
              leaf mt-id {
                type uint8;
                description
                  "The MT-ID for the topology enabled for the
                   external or NSSA prefix.";
              }
              leaf flags {
                type bits {
                  bit E {
                    description
                      "When set, the metric specified is a Type 2
                       external metric.";
                  }
                }
                description "Flags.";
              }
```

```
            leaf metric {
              type uint24;
              description "Metric for the topology.";
            }
            leaf forwarding-address {
              type inet:ipv4-address;
              description
                "Forwarding address.";
            }
            leaf external-route-tag {
              type uint32;
              description
                "Route tag for the topology.";
            }
          }
        }
      }
      container opaque {
        when "../../header/type = 9 or "
           + "../../header/type = 10 or "
           + "../../header/type = 11" {
          description
            "Only applies to Opaque LSAs.";
        }
        description
          "Opaque LSA.";

        uses unknown-tlvs;

        container node-tag-tlvs {
          description
            "All node tag tlvs.";
          list node-tag-tlv {
            description
              "Node tag tlv.";
            uses node-tag-tlv;
          }
        }

        container router-address-tlv {
          description
            "Router address TLV.";
          leaf router-address {
            type inet:ipv4-address;
            description
              "Router address.";
          }
        }
```

```
         container link-tlvs {
           description "All link TLVs in the LSA.";
           list link-tlv {
             description "Link TLV.";
             leaf link-type {
               type uint8;
               mandatory true;
               description "Link type.";
             }
             leaf link-id {
               type union {
                 type inet:ipv4-address;
                 type yang:dotted-quad;
               }
               mandatory true;
               description "Link ID.";
             }
             container local-if-ipv4-addrs {
               description "All local interface IPv4 addresses.";
               leaf-list local-if-ipv4-addr {
                 type inet:ipv4-address;
                 description
                   "List of local interface IPv4 addresses.";
               }
             }
             container remote-if-ipv4-addrs {
               description "All remote interface IPv4 addresses.";
               leaf-list remote-if-ipv4-addr {
                 type inet:ipv4-address;
                 description
                   "List of remote interface IPv4 addresses.";
               }
             }
             leaf te-metric {
               type uint32;
               description "TE metric.";
             }
             leaf max-bandwidth {
               type rt-types:bandwidth-ieee-float32;
               description "Maximum bandwidth.";
             }
             leaf max-reservable-bandwidth {
               type rt-types:bandwidth-ieee-float32;
               description "Maximum reservable bandwidth.";
             }
             container unreserved-bandwidths {
               description "All unreserved bandwidths.";
               list unreserved-bandwidth {
```

```
                 leaf priority {
                   type uint8 {
                     range "0 .. 7";
                   }
                   description "Priority from 0 to 7.";
                 }
                 leaf unreserved-bandwidth {
                   type rt-types:bandwidth-ieee-float32;
                   description "Unreserved bandwidth.";
                 }
                 description
                   "List of unreserved bandwidths for different
                    priorities.";
               }
             }
             leaf admin-group {
               type uint32;
               description
                 "Administrative group/Resource Class/Color.";
             }
             uses unknown-tlvs;
           }
         }

         container extended-prefix-tlvs {
           description "All extended prefix TLVs in the LSA.";
           list extended-prefix-tlv {
             description "Extended prefix TLV.";
             leaf route-type {
               type enumeration {
                 enum unspecified {
                   value "0";
                   description "Unspecified.";
                 }
                 enum intra-area {
                   value "1";
                   description "OSPF intra-area route.";
                 }
                 enum inter-area {
                   value "3";
                   description "OSPF inter-area route.";
                 }
                 enum external {
                   value "5";
                   description "OSPF External route.";
                 }
                 enum nssa {
                   value "7";
```

```
                  description "OSPF NSSA external route.";
                }
              }
              description "Route type.";
            }
            leaf flags {
              type bits {
                bit A {
                  description
                    "Attach flag.";
                }
                bit N {
                  description
                    "Node flag.";
                }
              }
              description "Flags.";
            }
            leaf prefix {
              type inet:ip-prefix;
              description "Address prefix.";
            }
            uses unknown-tlvs;
          }
        }

        container extended-link-tlvs {
          description "All extended link TLVs in the LSA.";
          list extended-link-tlv {
            description "Extended link TLV.";
            uses ospfv2-router-link;
            uses unknown-tlvs;
          }
        }
      }
    }

    grouping ospfv3-lsa-options {
      description "OSPFv3 LSA options";
      leaf options {
        type bits {
          bit AF {
            description
              "When set, the router supprts OSPFv3 AFs as in RFC5838.";
          }
          bit DC {
            description
              "When set, the router supports demand circuits.";
```

```
          }
          bit R {
            description
              "When set, the originator is an active router.";
          }
          bit N {
            description
              "If set, the router is attached to an NSSA";
          }
          bit E {
            description
              "This bit describes the way AS-external LSAs
               are flooded";
          }
          bit V6 {
            description
              "If clear, the router/link should be excluded
               from IPv6 routing calculaton";
          }
        }
        mandatory true;
        description "OSPFv3 LSA options.";
      }
    }

    grouping ospfv3-lsa-prefix {
      description
        "OSPFv3 LSA prefix.";

      leaf prefix {
        type inet:ip-prefix;
        description
          "Prefix.";
      }
      leaf prefix-options {
        type bits {
          bit NU {
            description
              "When set, the prefix should be excluded
               from IPv6 unicast calculations.";
          }
          bit LA {
            description
              "When set, the prefix is actually an IPv6 interface
               address of the Advertising Router.";
          }
          bit P {
            description
```

```
                "When set, the NSSA area prefix should be
                 translated to an AS External LSA and readvertised
                 by the translating NSSA Border Router.";
            }
            bit DN {
              description
                "When set, the inter-area-prefix LSA or
                 AS-external LSA prefix has been advertised as an
                 L3VPN prefix.";
            }
          }
          mandatory true;
          description "Prefix options.";
        }
      }

      grouping ospfv3-lsa-external {
        description
          "AS-External and NSSA LSA.";
        leaf metric {
          type uint24;
          description "Metric";
        }

        leaf flags {
          type bits {
            bit E {
              description
                "When set, the metric specified is a Type 2
                 external metric.";
            }
            bit F {
              description
                "When set, a Forwarding Address is included
                 in the LSA.";
            }
            bit T {
              description
                "When set, an External Route Tag is included
                 in the LSA.";
            }
          }
          description "Flags.";
        }

        leaf referenced-ls-type {
          type uint16;
          description "Referenced Link State type.";
```

```
       }

       uses ospfv3-lsa-prefix;

       leaf forwarding-address {
         type inet:ipv6-address;
         description
           "Forwarding address.";
       }

       leaf external-route-tag {
         type uint32;
         description
           "Route tag.";
       }
       leaf referenced-link-state-id {
         type uint32;
         description
           "Referenced Link State ID.";
       }
     }

     grouping ospfv3-lsa-body {
       description "OSPFv3 LSA body.";
       container router {
         when "../../header/type = 8193" { // 0x2001
           description
             "Only applies to Router LSAs.";
         }
         description "Router LSA.";
         leaf flags {
           type bits {
             bit V {
               description
                 "When set, the router is an endpoint of one or
                  more virtual links.";
             }
             bit E {
               description
                 "When set, the router is an AS Boundary Router
                  (ASBR).";
             }
             bit B {
               description
                 "When set, the router is an Area Border
                  Router (ABR).";
             }
             bit Nt {
```

```
              description
                "When set, the router is an NSSA border router
                 that is unconditionally translating NSSA LSAs
                 into AS-external LSAs.";
            }
          }
          mandatory true;
          description "Router LSA flags.";
        }

        uses ospfv3-lsa-options;

        container links {
          description "All router link.";
          list link {
            description "Router LSA link.";
            leaf interface-id {
              type uint32;
              description "Interface ID.";
            }
            leaf neighbor-interface-id {
              type uint32;
              description "Neighbor Interface ID.";
            }
            leaf neighbor-router-id {
              type rt-types:router-id;
              description "Neighbor Router ID.";
            }
            leaf type {
              type uint8;
              description "Link type.";
            }
            leaf metric {
              type uint16;
                description "Metric.";
            }
          }
        }
      }
      container network {
        when "../../header/type = 8194" { // 0x2002
          description
            "Only applies to Network LSA.";
        }
        description "Network LSA.";

        uses ospfv3-lsa-options;
```

```
         container attached-routers {
           description "All attached routers.";
           leaf-list attached-router {
             type yang:dotted-quad;
             description
               "List of the routers attached to the network.";
           }
         }
       }
       container inter-area-prefix {
         when "../../header/type = 8195" { // 0x2003
           description
             "Only applies to Inter-Area-Prefix LSAs.";
         }
         leaf metric {
           type uint24;
           description "Metric";
         }
         uses ospfv3-lsa-prefix;
         description "Inter-Area-Prefix LSA.";
       }
       container inter-area-router {
         when "../../header/type = 8196" { // 0x2004
           description
             "Only applies to Inter-Area-Router LSAs.";
         }
         uses ospfv3-lsa-options;
         leaf metric {
           type uint24;
           description "Metric.";
         }
         leaf destination-router-id {
           type rt-types:router-id;
           description
             "The Router ID of the router being described by the LSA.";
         }
         description "Inter-Area-Router LSA.";
       }
       container as-external {
         when "../../header/type = 16389" { // 0x4005
           description
             "Only applies to AS-external LSAs.";
         }

         uses ospfv3-lsa-external;

         description "AS-External LSA.";
       }
```

```
       container nssa {
         when "../../header/type = 8199" { // 0x2007
           description
             "Only applies to NSSA LSAs.";
         }
         uses ospfv3-lsa-external;

         description "NSSA LSA.";
       }
       container link {
         when "../../header/type = 8" { // 0x0008
           description
             "Only applies to Link LSAs.";
         }
         leaf rtr-priority {
           type uint8;
           description "Router Priority for the interface.";
         }
         uses ospfv3-lsa-options;

         leaf link-local-interface-address {
           type inet:ipv6-address;
           description
             "The originating router’s link-local
              interface address for the link.";
         }

         leaf num-of-prefixes {
           type uint32;
           description "Number of prefixes.";
         }

         container prefixes {
           description "All prefixes for the link.";
           list prefix {
             description
               "List of prefixes associated with the link.";
             uses ospfv3-lsa-prefix;
           }
         }
         description "Link LSA.";
       }
       container intra-area-prefix {
         when "../../header/type = 8201" { // 0x2009
           description
             "Only applies to Intra-Area-Prefix LSA.";
         }
         description "Intra-Area-Prefix LSA.";
```

```
          leaf referenced-ls-type {
            type uint16;
            description "Referenced Link State type.";
          }
          leaf referenced-link-state-id {
            type uint32;
            description
              "Referenced Link State ID.";
          }
          leaf referenced-adv-router {
            type inet:ipv4-address;
            description
              "Referenced Advertising Router.";
          }

          leaf num-of-prefixes {
            type uint16;
            description "Number of prefixes.";
          }
          container prefixes {
            description "All prefixes in this LSA.";
            list prefix {
              description "List of prefixes in this LSA.";
              uses ospfv3-lsa-prefix;
              leaf metric {
                type uint24;
                description "Prefix Metric.";
              }
            }
          }
        }
        container router-information {
          when "../../header/type = 32780 or " // 0x800C
             + "../../header/type = 40972 or " // 0xA00C
             + "../../header/type = 49164 or " // 0xC00C
             + "../../header/type = 57356" {   // 0xE00C
            description
              "Only applies to Router Information LSAs (RFC7770).";
          }
          container node-tag-tlvs {
            description
              "All node tag tlvs.";
            list node-tag-tlv {
              description
                "Node tag tlv.";
              uses node-tag-tlv;
            }
          }
```

```
         description "Router Information LSA.";
       }
     }

     grouping lsa-header {
       description
           "Common LSA for OSPFv2 and OSPFv3";
       leaf age {
         type uint16;
         mandatory true;
         description "LSA age.";
       }
       leaf type {
         type uint16;
         mandatory true;
         description "LSA type.";
       }
       leaf adv-router {
         type yang:dotted-quad;
         mandatory true;
         description "LSA advertising router.";
       }
       leaf seq-num {
         type uint32;
         mandatory true;
         description "LSA sequence number.";
       }
       leaf checksum {
         type checksum16-type;
         mandatory true;
         description "LSA checksum.";
       }
       leaf length {
         type uint16;
         mandatory true;
         description "LSA length.";
       }
     }

     grouping ospfv2-lsa {
       description
           "OSPFv2 LSA.";
       container header {
         must "(type = 9 or type = 10 or type = 11) and "
            + "opaque-id and opaque-type "
            + "or (type != 9 and type != 10 and type != 11) "
            + "and not(opaque-id) and "
            + "not(opaque-type)" {
```

```
          description
            "Opaque type and ID only apply to Opaque LSAs.";
        }
        description
          "Decoded OSPFv2 LSA header data.";
        leaf option {
          type bits {
            bit MT {
              description
                "When set, the router supports multi-topology as
                 in RFC 4915.";
            }
            bit DC {
              description
                "When set, the router supports demand circuits.";
            }
            bit P {
              description
                "Only used in type-7 LSA. When set, an NSSA
                 border router should translate the type-7 LSA
                 to a type-5 LSA.";
            }
            bit MC {
              description
                "When set, the router supports MOSPF.";
            }
            bit E {
              description
                "This bit describes the way AS-external LSAs
                 are flooded.";
            }
            bit O {
              description
                "When set, the router is opaque-capable as in
                 RFC 5250.";
            }
            bit DN {
              description
                "When a type 3, 5 or 7 LSA is sent from a PE to a CE,
                 the DN bit must be set. See RFC 4576.";
            }
          }
          mandatory true;
          description "LSA options.";
        }
        leaf lsa-id {
          type yang:dotted-quad;
          mandatory true;
```

```
          description "LSA ID.";
        }

        leaf opaque-type {
          type uint8;
          description "Opaque type.";
        }

        leaf opaque-id {
          type uint24;
          description "Opaque ID.";
        }

        uses lsa-header;
      }
      container body {
        description
          "Decoded OSPFv2 LSA body data.";
        uses ospfv2-lsa-body;
      }
    }

    grouping ospfv3-lsa {
      description
          "Decoded OSPFv3 LSA.";
      container header {
        description
          "Decoded OSPFv3 LSA header data.";
        leaf lsa-id {
          type uint32;
          mandatory true;
          description "LSA ID.";
        }
        uses lsa-header;
      }
      container body {
        description
          "Decoded OSPF LSA body data.";
        uses ospfv3-lsa-body;
      }
    }
    grouping lsa-common {
      description
          "Common field for OSPF LSA represenation.";
      leaf decoded-completed {
        type boolean;
        description
          "The OSPF LSA body is fully decoded.";
```

```
       }
       leaf raw-data {
         type yang:hex-string;
         description
           "The complete LSA in network byte
            order hexadecimal as received or originated.";
       }
     }

     grouping lsa {
       description
           "OSPF LSA.";
       uses lsa-common;
       choice version {
         description
           "OSPFv2 or OSPFv3 LSA body.";
         container ospfv2 {
           description "OSPFv2 LSA";
           uses ospfv2-lsa;
         }
         container ospfv3 {
           description "OSPFv3 LSA";
           uses ospfv3-lsa;
         }
       }
     }

     grouping lsa-key {
       description
         "OSPF LSA key.";
       leaf lsa-id {
         type union {
           type yang:dotted-quad;
           type uint32;
         }
         description
           "LSA ID.";
       }
       leaf adv-router {
         type inet:ipv4-address;
         description
           "Advertising router.";
       }
     }

     grouping instance-stat {
       description "Per-instance statistics";
       leaf originate-new-lsa-count {
```

```
         type yang:counter32;
         description "The number of new LSAs originated.";
       }
       leaf rx-new-lsas-count {
         type yang:counter32;
         description "The number of LSAs received.";
       }
       leaf as-scope-lsa-count {
         type yang:gauge32;
         description "The number of AS Scope LSAs.";
       }
       leaf as-scope-lsa-chksum-sum {
         type uint32;
         description
           "The sum of the LSA checksums for AS Scope LSAs.";
       }
       container database {
         description "Container for per AS-scope LSA statistics.";
         list as-scope-lsa-type {
           description "List of AS-scope LSA statistics";
           leaf lsa-type {
             type uint16;
             description "AS-scope LSA type.";
           }
           leaf lsa-count {
             type yang:gauge32;
             description "The number of LSAs of the LSA type.";
           }
           leaf lsa-cksum-sum {
             type int32;
             description
               "The sum of the LSA checksums of the LSA type.";
           }
         }
       }
     }

     grouping area-stat {
       description "Per-area statistics.";
       leaf spf-runs-count {
         type yang:counter32;
         description
           "The number of times the intra-area SPF has run.";
       }
       leaf abr-count {
         type yang:gauge32;
         description
           "The total number of Area Border Routers (ABRs)
```

```
             reachable within this area.";
        }
        leaf asbr-count {
          type yang:gauge32;
          description
            "The total number of AS Border Routers (ASBRs).";
        }
        leaf ar-nssa-translator-event-count {
          type yang:counter32;
          description
            "The number of NSSA translator-state changes.";
        }
        leaf area-scope-lsa-count {
          type yang:gauge32;
          description
            "The number of area scope LSAs in the area.";
        }
        leaf area-scope-lsa-cksum-sum {
          type int32;
          description "The sum of the area scope LSAs checksums.";
        }
        container database {
          description "Container for area scope LSA type statistics.";
          list area-scope-lsa-type {
            description "List of area scope LSA statistics";
            leaf lsa-type {
              type uint16;
              description "Area scope LSA type.";
            }
            leaf lsa-count {
              type yang:gauge32;
              description "The number of LSAs of the LSA type.";
            }
            leaf lsa-cksum-sum {
              type int32;
              description
                "The sum of the LSA checksums of the LSA type.";
            }
          }
        }
      }

      grouping interface-stat {
        description "Per-interface statistics";
        leaf if-event-count {
          type yang:counter32;
          description
            "The number of times this interface has changed its
```

```
             state or an error has occurred.";
         }
         leaf link-scope-lsa-count {
           type yang:gauge32;
           description "The number of link scope LSAs.";
         }
         leaf link-scope-lsa-cksum-sum {
           type uint32;
           description "The sum of link scope LSA checksums.";
         }
         container database {
           description "Container for link scope LSA type statistics.";
           list link-scope-lsa-type {
             description "List of link scope LSA statistics";
             leaf lsa-type {
               type uint16;
               description "Link scope LSA type.";
             }
             leaf lsa-count {
               type yang:gauge32;
               description "The number of LSAs of the LSA type.";
             }
             leaf lsa-cksum-sum {
               type int32;
               description
                 "The sum of the LSA checksums of the LSA type.";
             }
           }
         }
       }

       grouping neighbor-stat {
         description "Per-neighbor statistics.";
         leaf nbr-event-count {
           type yang:counter32;
           description
             "The number of times this neighbor has changed
              state or an error has occurred.";
         }
         leaf nbr-retrans-qlen {
           type yang:gauge32;
           description
             "The current length of the retransmission queue.";
         }
       }

       grouping instance-fast-reroute-config {
         description
```

```
         "This group defines global configuration of IP-FRR.";
       container fast-reroute {
         if-feature fast-reroute;
         description
           "This container may be augmented with global
            parameters for IP-FRR.";
         container lfa {
           if-feature lfa;
           description
             "This container may be augmented with
              global parameters for LFA. Container creation
              has no effect on LFA activation.";
         }
       }
     }

     grouping interface-fast-reroute-config {
       description
         "This group defines interface configuration of IP-FRR.";
       container fast-reroute {
         if-feature fast-reroute;
         container lfa {
           if-feature lfa;
           leaf candidate-enable {
             type boolean;
             default true;
             description
               "Enable the interface to be used as backup.";
           }
           leaf enable {
             type boolean;
             default false;
             description
               "Activates LFA - Per-prefix LFA computation
                is assumed.";
           }
           container remote-lfa {
             if-feature remote-lfa;
             leaf enable {
               type boolean;
               default false;
               description
               "Activates Remote LFA (R-LFA).";
             }
             description
               "Remote LFA configuration.";
           }
           description
```

```
              "LFA configuration.";
          }
          description
            "IP Fast-reroute configuration.";
        }
      }

      grouping interface-physical-link-config {
        description
          "Interface cost configuration that only applies to
           physical interfaces and sham links.";
        leaf cost {
          type uint16 {
            range "1..65535";
          }
          description
            "Interface cost.";
        }
        leaf mtu-ignore {
          if-feature mtu-ignore;
          type boolean;
          description
            "Enable/Disable bypassing the MTU mismatch check in
             Database Description packets.";
        }
        leaf prefix-suppression {
          if-feature prefix-suppression;
          type boolean;
          description
            "Suppress advertisement of the prefixes associated
             with the interface.";
        }
      }

      grouping interface-common-config {
        description
          "Common configuration for all types of interfaces,
           including virtual links and sham links.";

        leaf hello-interval {
          type uint16 {
            range "1..65535";
          }
          units seconds;
          description
            "Interval between hello packets in seconds.";
        }
```

```
       leaf dead-interval {
         type uint32 {
           range "1..2147483647";
         }
         units seconds;
         must "../dead-interval > ../hello-interval" {
           error-message "The dead interval must be "
                       + "larger than the hello interval";
           description
             "The value MUST be greater than 'hello-internval'.";
         }
         description
           "Interval after which a neighbor is
            declared down in seconds.";
       }

       leaf retransmit-interval {
         type uint16 {
           range "1..3600";
         }
         units seconds;
         description
           "Interval between retransmitting unacknowledged Link
            State Advertisements (LSAs) in seconds.";
       }

       leaf transmit-delay {
         type uint16 {
           range "1..3600";
         }
         units seconds;
         description
           "Estimated time needed to transmit Link State Update
            packets on the interface in seconds.";
       }

       leaf lls {
         if-feature lls;
         type boolean;
         description
           "Enable/Disable link-local signaling (LLS) support.";
       }

       container ttl-security {
         if-feature ttl-security;
         description "TTL security check.";
         leaf enable {
           type boolean;
```

```
            description
              "Enable/Disable TTL security check.";
          }
          leaf hops {
            type uint8 {
              range "1..254";
            }
            description
              "Maximum number of hops that an OSPF packet may
               have traversed before reception.";
          }
        }
        leaf enable {
          if-feature admin-control;
          type boolean;
          default true;
          description
            "Enable/disable protocol on the interface.";
        }

        container authentication {
          description "Authentication configuration.";
          choice auth-type-selection {
            description
              "Options for expressing authentication setting.";
            case auth-ipsec {
              when "../../../../../../../rt:type = 'ospf:ospfv3'" {
                description "Applied to OSPFv3 only.";
              }
              if-feature ospfv3-authentication-ipsec;
              leaf sa {
                  type string;
                  description
                    "Security Association name.";
              }
            }
            case auth-trailer-key-chain {
              leaf key-chain {
                type key-chain:key-chain-ref;
                 description
                  "key-chain name.";
                }
              }
            case auth-trailer-key {
              leaf key {
                type string;
                description
                  "Key string in ASCII format.";
```

```
            }
            leaf crypto-algorithm {
              type identityref {
                base key-chain:crypto-algorithm;
              }
              description
                "Cryptographic algorithm associated with key.";
            }
          }
        }
      }
    } // interface-common-config

    grouping interface-config {
      description "Configuration for real interfaces.";

      leaf interface-type {
        type enumeration {
          enum "broadcast" {
            description
              "Specify OSPF broadcast multi-access network.";
          }
          enum "non-broadcast" {
            description
              "Specify OSPF Non-Broadcast Multi-Access
               (NBMA) network.";
          }
          enum "point-to-multipoint" {
            description
              "Specify OSPF point-to-multipoint network.";
          }
          enum "point-to-point" {
            description
              "Specify OSPF point-to-point network.";
          }
        }
        description
          "Interface type.";
      }

      leaf passive {
        type boolean;
        description
          "Enable/Disable passive interface - a passive interface's
           prefix will be advertised but no neighbor adjacencies
           will be formed on the interface.";
      }
```

```
       leaf demand-circuit {
         if-feature demand-circuit;
         type boolean;
         description
           "Enable/Disable demand circuit.";
       }

       leaf priority {
         type uint8;
         description
           "Configure OSPF router priority.";
       }

       container multi-areas {
         if-feature multi-area-adj;
         description "Container for multi-area config.";
         list multi-area {
           key multi-area-id;
           description
             "Configure OSPF multi-area adjacency.";
           leaf multi-area-id {
             type area-id-type;
             description
               "Multi-area adjacency area ID.";
           }
           leaf cost {
             type uint16;
             description
               "Interface cost for multi-area adjacency.";
           }
         }
       }

       container static-neighbors {
         description "Statically configured neighbors.";

         list neighbor {
           key "identifier";
           description
             "Specify a static OSPF neighbor.";

           leaf identifier {
             type inet:ip-address;
             description
               "Neighbor router ID, IPv4 address, or IPv6 address.";
           }

           leaf cost {
```

```
            type uint16 {
              range "1..65535";
            }
            description "Neighbor cost.";
          }
          leaf poll-interval {
            type uint16 {
              range "1..65535";
            }
            units seconds;
            description "Neighbor poll interval in seconds.";
          }
          leaf priority {
            type uint8 {
              range "1..255";
            }
            description "Neighbor priority for DR election.";
          }
        }
      }

      leaf node-flag {
        if-feature node-flag;
        type boolean;
        default false;
        description
          "Set prefix as a node representative prefix.";
      }

      container bfd {
        if-feature bfd;
        description "BFD configuration.";
        leaf enable {
          type boolean;
          default false;
          description
            "True if BFD is enabled for the OSPF interface.";
        }
      }

      uses interface-fast-reroute-config;
      uses interface-common-config;
      uses interface-physical-link-config;
    } // grouping interface-config

    grouping neighbor-state {
      description
        "OSPF neighbor operational state.";
```

```
        leaf address {
          type inet:ip-address;
          config false;
          description
            "Neighbor address.";
        }
        leaf dr-router-id {
          type rt-types:router-id;
          config false;
          description "Neighbor’s Designated Router (DR) router ID.";
        }

        leaf dr-ip-addr {
          type inet:ip-address;
          config false;
          description "Neighbor’s Designated Router (DR) IP address.";
        }

        leaf bdr-router-id {
          type rt-types:router-id;
          config false;
          description
            "Neighbor’s Backup Designated Router (BDR) router ID.";
        }

        leaf bdr-ip-addr {
          type inet:ip-address;
          config false;
          description
            "Neighbor’s Backup Designated Router (BDR) IP Address.";
        }
        leaf state {
          type nbr-state-type;
          config false;
          description
            "OSPF neighbor state.";
        }
        leaf dead-timer {
          type uint32;
          units "seconds";
          config false;
          description "This timer tracks the remaining time before
                       the neighbor is declared dead.";
        }
        container statistics {
          config false;
          description "Per neighbor statistics";
          uses neighbor-stat;
```

```
      }
    }

    grouping interface-common-state {
      description
        "OSPF interface common operational state.";
      reference "RFC2328 Section 9";

      leaf state {
        type if-state-type;
        config false;
        description "Interface state.";
      }

      leaf hello-timer {
        type uint32;
        units "seconds";
        config false;
        description "This timer tracks the remaining time before
                     the next hello packet is sent.";
      }

      leaf wait-timer {
        type uint32;
        units "seconds";
        config false;
        description "This timer tracks the remaining time before
                     the interface exits the Waiting state.";
      }

      leaf dr-router-id {
        type rt-types:router-id;
        config false;
        description "Designated Router (DR) router ID.";
      }

      leaf dr-ip-addr {
        type inet:ip-address;
        config false;
        description "Designated Router (DR) IP address.";
      }

      leaf bdr-router-id {
        type rt-types:router-id;
        config false;
        description "Backup Designated Router (BDR) router ID.";
      }
```

```
       leaf bdr-ip-addr {
         type inet:ip-address;
         config false;
         description "Backup Designated Router (BDR) IP Address.";
       }

       container statistics {
         config false;
         description "Per interface statistics";
         uses interface-stat;
       }

       container neighbors {
         config false;
         description "All neighbors for the interface.";
         list neighbor {
           key "neighbor-router-id";
           description
             "List of OSPF neighbors.";
           leaf neighbor-router-id {
             type rt-types:router-id;
             description
               "Neighbor router ID.";
           }
           uses neighbor-state;
         } // list of OSPF neighbors
       }
       container database {
         config false;
         description "Link scope LSA database.";
         list link-scope-lsa-type {
           key "lsa-type";
           description
             "List OSPF link scope LSA databases.";
           leaf lsa-type {
             type uint16;
             description "OSPF link scope LSA type.";
           }
           container link-scope-lsas {
             description
               "All link scope LSAs of this LSA type.";
             list link-scope-lsa {
               key "lsa-id adv-router";
               description "List of OSPF link scope LSAs";
               uses lsa-key;
               uses lsa {
                 refine "version/ospfv2/ospfv2" {
                   must "../../../../../../../../../../../"
```

```
                     + "rt:type = 'ospf:ospfv2'" {
                    description "OSPFv2 LSA.";
                  }
                }
                refine "version/ospfv3/ospfv3" {
                  must "../../../../../../../../../../../"
                     + "rt:type = 'ospf:ospfv3'" {
                    description "OSPFv3 LSA.";
                  }
                }
              }
            }
          }
        } // list link-scope-lsas
      }
    } // interface-common-state

    grouping interface-state {
      description
        "OSPF interface operational state.";
      reference "RFC2328 Section 9";

      uses interface-common-state;
    }

    grouping virtual-link-config {
      description
        "OSPF virtual link configuration state.";

      uses interface-common-config;
    }

    grouping virtual-link-state {
      description
        "OSPF virtual link operational state.";

      leaf cost {
        type uint16 {
          range "1..65535";
        }
        config false;
        description
          "Virtual link interface cost.";
      }
      uses interface-common-state;
    }

    grouping sham-link-config {
```

```
        description
          "OSPF sham link configuration state.";

        uses interface-common-config;
        uses interface-physical-link-config;
      }

      grouping sham-link-state {
        description
          "OSPF sham link operational state.";
        /* All container/leaf should be config false. */
        uses interface-common-state;
      }

      grouping af-area-config {
        description
          "OSPF address-family specific area config state.";

        container ranges {
          description "Container for summary ranges";

          list range {
            key "prefix";
            description
              "Summarize routes matching address/mask
               (Area Border Routers (ABRs) only)";
            leaf prefix {
              type inet:ip-prefix;
              description
                "IPv4 or IPv6 prefix";
            }
            leaf advertise {
              type boolean;
              description
                "Advertise or hide.";
            }
            leaf cost {
              type uint24 {
                range "0..16777214";
              }
              description
                "Advertised cost of summary route.";
            }
          }
        }
      }

      grouping area-common-config {
```

```
       description
         "OSPF area common configuration state.";

       leaf summary {
         when "../area-type = 'ospf:stub' or "
            + "../area-type = 'ospf:nssa'" {
           description
             "Summary advertisement into the stub/NSSA area.";
         }
         type boolean;
         description
           "Enable/Disable summary advertisement into the stub or
            NSSA area.";
       }
       leaf default-cost {
         when "../area-type = 'ospf:stub' or "
            + "../area-type = 'ospf:nssa'" {
             description
               "Cost for LSA default route advertised into the
                stub or NSSA area.";
         }
         type uint32 {
           range "1..16777215";
         }
         description
           "Set the summary default route cost for a
            stub or NSSA area.";
       }
     }

     grouping area-config {
       description
         "OSPF area configuration state.";

       leaf area-type {
         type identityref {
           base area-type;
         }
         default normal;
         description
           "Area type.";
       }

       uses area-common-config;

       uses af-area-config {
         when "../../../operation-mode = "
            + "'ospf:ships-in-the-night'" {
```

```
          description
            "Ships in the night configuration.";
        }
      }
    }

    grouping area-state {
      description
        "OSPF area operational state.";

      container statistics {
        config false;
        description "Per area statistics";
        uses area-stat;
      }

      container database {
        config false;
        description "Area scope LSA database.";
        list area-scope-lsa-type {
          key "lsa-type";
          description "List OSPF area scope LSA databases.";
          leaf lsa-type {
            type uint16;
            description "OSPF area scope LSA type.";
          }
          container area-scope-lsas {
            description
              "All area scope LSAs of an area scope
               LSA type.";
            list area-scope-lsa {
              key "lsa-id adv-router";
              description "List of OSPF area scope LSAs";
              uses lsa-key;
              uses lsa {
                refine "version/ospfv2/ospfv2" {
                  must "../../../../../../../../../../"
                     + "rt:type = 'ospf:ospfv2'" {
                    description "OSPFv2 LSA.";
                  }
                }
                refine "version/ospfv3/ospfv3" {
                  must "../../../../../../../../../../"
                     + "rt:type = 'ospf:ospfv3'" {
                    description "OSPFv3 LSA.";
                  }
                }
              }
```

```
             }
           }
         } // list area-scope-lsas
       }
     }

     grouping local-rib {
       description "Local-rib grouping.";
       container local-rib {
         config false;
         description "Local-rib.";
         list route {
           key "prefix";
           description "Routes";
           leaf prefix {
             type inet:ip-prefix;
             description "Destination prefix.";
           }
           container next-hops {
             description "All next hops for the route.";
             list next-hop {
               key "next-hop";
               description "List of next hop for the route";
               leaf outgoing-interface {
                 type if:interface-ref;
                 description
                   "Name of the outgoing interface.";
               }
               leaf next-hop {
                type inet:ip-address;
                description "Nexthop address.";
               }
             }
           }
           leaf metric {
             type uint32;
             description "Metric for this route.";
           }
           leaf route-type {
             type route-type;
             description "Route type for this route.";
           }
           leaf route-tag {
             type uint32;
             description "Route tag for this route.";
           }
         }
       }
```

```
     }

     grouping ietf-spf-delay-config {
       leaf initial-delay {
         type uint16;
         units msec;
         description
           "Delay used while in QUIET state.";
       }
       leaf short-delay {
         type uint16;
         units msec;
         description
           "Delay used while in SHORT_WAIT state.";
       }
       leaf long-delay {
         type uint16;
         units msec;
         description
           "Delay used while in LONG_WAIT state.";
       }
       leaf hold-down {
         type uint16;
         units msec;
         description
           "Timer used to consider an IGP stability period.";
       }
       leaf time-to-learn {
         type uint16;
         units msec;
         description
           "Duration used to learn all the IGP events
            related to a single component failure.";
       }
       description
         "Grouping for IETF SPF delay configuration.";
     }

     grouping ietf-spf-delay-state {
       leaf current-state {
         type enumeration {
           enum "QUIET" {
             description "QUIET state";
           }
           enum "SHORT_WAIT" {
             description "SHORT_WAIT state";
           }
           enum "LONG_WAIT" {
```

```
             description "LONG_WAIT state";
           }
         }
         config false;
         description
           "Current state of the algorithm.";
       }
       leaf remaining-time-to-learn {
         type uint16;
         units "seconds";
         config false;
         description
           "Remaining time until time-to-learn timer fires.";
       }
       leaf remaining-hold-down {
         type uint16;
         units "seconds";
         config false;
         description
           "Remaining time until hold-down timer fires.";
       }
       leaf last-event-received {
         type yang:timestamp;
         config false;
         description
           "Time of last IGP event received";
       }
       leaf next-spf-time {
         type yang:timestamp;
         config false;
         description
           "Time when next SPF has been scheduled.";
       }
       leaf last-spf-time {
         type yang:timestamp;
         config false;
         description
           "Time of last SPF computation.";
         }
       description
         "Grouping for IETF SPF delay operational states.";
     }

     grouping node-tag-config {
       description
         "OSPF node tag config state.";
       container node-tags {
         if-feature node-tag;
```

```
        list node-tag {
          key tag;
          leaf tag {
            type uint32;
              description
                "Node tag value.";
          }
          description
            "List of tags.";
        }
        description
          "Container for node tags.";
      }
    }

    grouping instance-config {
      description
        "OSPF instance config state.";

      leaf explicit-router-id {
        if-feature explicit-router-id;
        type rt-types:router-id;
        description
          "Defined in RFC 2328. A 32-bit number
           that uniquely identifies the router.";
      }

      container preference {
        description "Route preference config state.";
        choice scope {
          description
            "Options for expressing preference
             as single or multiple values.";
          case single-value {
            leaf all {
              type uint8;
              description
                "Preference for intra-area, inter-area and
                 external routes.";
            }
          }
          case multi-values {
            choice granularity {
              description
                "Options for expressing preference
                 for intra-area and inter-area routes.";
              case detail {
                leaf intra-area {
```

```
                  type uint8;
                  description
                    "Preference for intra-area routes.";
                }
                leaf inter-area {
                  type uint8;
                  description
                    "Preference for inter-area routes.";
                }
              }
              case coarse {
                leaf internal {
                  type uint8;
                  description
                    "Preference for both intra-area and
                     inter-area routes.";
                }
              }
            }
            leaf external {
              type uint8;
              description
                "Preference for external routes.";
            }
          }
        }
      }

      container nsr {
        if-feature nsr;
        description
          "Non-Stop Routing (NSR) config state.";
        leaf enable {
          type boolean;
          description
            "Enable/Disable NSR.";
        }
      }

      container graceful-restart {
        if-feature graceful-restart;
        description
          "Graceful restart config state.";
        leaf enable {
          type boolean;
          description
            "Enable/Disable graceful restart as defined in RFC 3623
             for OSPFv2 and RFC 5187 for OSPFv3.";
```

```
         }
         leaf helper-enable {
           type boolean;
           description
             "Enable graceful restart helper support for restarting
              routers (RFC 3623 Section 3).";
         }
         leaf restart-interval {
           type uint16 {
             range "1..1800";  // Range is defined in RFC 3623.
           }
           units seconds;
           default "120";  // Default is defined in RFC 3623.
           description
             "Interval in seconds to attempt graceful restart prior
              to failing (RFC 3623 Section B.1)";
         }
         leaf helper-strict-lsa-checking {
           type boolean;
           description
             "Terminate graceful restart when an LSA topology change
              is detected (RFC 3623 Section B.2).";
         }
       }

       leaf enable {
         if-feature admin-control;
         type boolean;
         default true;
         description
           "Enable/Disable the protocol.";
       }

       container auto-cost {
         if-feature auto-cost;
         description
           "Interface Auto-cost configuration state.";
         leaf enable {
           type boolean;
           description
             "Enable/Disable interface auto-cost.";
         }
         leaf reference-bandwidth {
           when "../enable = 'true'" {
             description "Only when auto cost is enabled";
           }
           type uint32 {
             range "1..4294967";
```

```
          }
          units Mbits;
          description
            "Configure reference bandwidth used to automatically
             determine interface cost (Mbits). The cost is the
             reference bandwidth divided by the interface speed
             with 1 being the minimum cost.";
        }
      }

      container spf-control {
        leaf paths {
          if-feature max-ecmp;
          type uint16 {
            range "1..32";
          }
          description
            "Maximum number of Equal-Cost Multi-Path (ECMP) paths.";
        }
        container ietf-spf-delay {
          if-feature ietf-spf-delay;
          uses ietf-spf-delay-config;
          description
            "IETF spf delay algorithm configuration.";
        }
        description "SPF calculation control.";
      }

      container database-control {
        leaf max-lsa {
          if-feature max-lsa;
          type uint32 {
            range "1..4294967294";
          }
          description
            "Maximum number of LSAs OSPF the router will accept.";
        }
        description "Database maintenance control.";
      }

      container stub-router {
        if-feature stub-router;
        description "Set maximum metric configuration";

        choice trigger {
          description
            "Specific triggers which will enable stub
             router state.";
```

```
          container always {
            presence
              "Enables unconditional stub router support";
            description
              "Unconditional stub router state (advertise
               transit links with max metric";
          }
        }
      }

      container mpls {
        description
          "OSPF MPLS config state.";
        container te-rid {
          if-feature te-rid;
          description
            "Stable OSPF Router IP Address used for Traffic
             Engineering (TE)";
          leaf ipv4-router-id {
            type inet:ipv4-address;
            description
              "Explicitly configure the TE IPv4 router ID.";
          }
          leaf ipv6-router-id {
            type inet:ipv6-address;
            description
              "Explicitly configure the TE IPv6 router ID.";
          }
        }
        container ldp {
          description
            "OSPF MPLS LDP config state.";
          leaf igp-sync {
            if-feature ldp-igp-sync;
            type boolean;
            description
              "Enable LDP IGP synchronization.";
          }
        }
      }
      uses instance-fast-reroute-config;
      uses node-tag-config;
    }

    grouping instance-state {
      description
        "OSPF instance operational state.";
```

```
        leaf router-id {
          type rt-types:router-id;
          config false;
          description
            "Defined in RFC 2328. A 32-bit number
             that uniquely identifies the router.";
        }

        uses local-rib;

        container statistics {
          config false;
          description "Per instance statistics";
          uses instance-stat;
        }

        container ietf-spf-delay {
          if-feature ietf-spf-delay;
          config false;
          uses ietf-spf-delay-config;
          uses ietf-spf-delay-state;
          description
            "IETF SPF delay operational states.";
        }

        container database {
          config false;
          description "AS scope LSA database.";
          list as-scope-lsa-type {
            key "lsa-type";
            description "List OSPF AS scope LSA databases.";
            leaf lsa-type {
              type uint16;
              description "OSPF AS scope LSA type.";
            }
            container as-scope-lsas {
              description "All AS scope of LSA of this LSA type.";
              list as-scope-lsa {
                key "lsa-id adv-router";
                description "List of OSPF area scope LSAs";
                uses lsa-key;
                uses lsa {
                  refine "version/ospfv2/ospfv2" {
                    must "../../../../../../../rt:type = "
                       + "'ospf:ospfv2'" {
                      description "OSPFv2 LSA.";
                    }
                  }
```

```
                 refine "version/ospfv3/ospfv3" {
                   must "../../../../../../../../rt:type = "
                      + "'ospf:ospfv3'" {
                     description "OSPFv3 LSA.";
                   }
                 }
               }
             }
           }
         } // list as-scope-lsas
       }
       uses spf-log;
       uses lsa-log;
     }

     grouping ospf-config {
       description
         "OSPF top configuration state.";

       leaf operation-mode {
         type identityref {
           base operation-mode;
         }
         default ospf:ships-in-the-night;
         description
           "OSPF operation mode.";
       }
     }

     grouping ospf-state {
       /* All leaf/container must be config false. */
       description
         "OSPF top operational state.";
     }

     grouping multi-topology-area-common-config {
       description
         "OSPF multi-topology area common configuration state.";
       leaf summary {
         when "../../../../../areas/area[area-id=current()/../area-id]/"
            + "area-type = 'ospf:stub' or "
            + "../../../../../areas/area[area-id=current()/../area-id]/"
            + "area-type = 'ospf:nssa'" {
           description
             "Summary advertisement into the stub/NSSA area.";
         }
         type boolean;
         description
```

```
            "Enable/Disable summary advertisement into the
             topology in the stub or NSSA area.";
        }
        leaf default-cost {
          when "../../../../../areas/area[area-id=current()/../area-id]/"
             + "area-type = 'ospf:stub' or "
             + "../../../../../areas/area[area-id=current()/../area-id]/"
             + "area-type = 'ospf:nssa'" {
            description
              "Cost for LSA default route advertised into the
               topology into the stub or NSSA area.";
          }
          type uint32 {
            range "1..16777215";
          }
          description
            "Set the summary default route cost for a
             stub or NSSA area.";
        }
      }

      grouping multi-topology-area-config {
        description
          "OSPF multi-topology area configuration state.";

        uses multi-topology-area-common-config;

        uses af-area-config {
          when "../../../../../operation-mode = "
             + "'ospf:ships-in-the-night'" {
            description
              "Ships in the night configuration.";
          }
        }
      }

      grouping multi-topology-area-state {
        /* All leaf/container must be config false. */
        description
          "OSPF multi-topology area operational state.";
      }

      grouping multi-topology-config {
        description
          "OSPF multi-topology configuration state.";
      }

      grouping multi-topology-state {
```

```
       /* All leaf/container must be config false. */
       description
         "OSPF multi-topology operational state.";

       uses local-rib;
     }

     grouping multi-topology-interface-config {
       description
         "OSPF multi-topology configuration state.";

       leaf cost {
          type uint32;
          description
            "Interface cost for this topology.";
       }
     }

     grouping multi-topology-interface-state {
       /* All leaf/container must be config false. */
       description
         "OSPF multi-topology operational state.";
     }

     grouping ospfv3-interface-config {
       description
         "OSPFv3 interface specific configuration state.";

       leaf instance-id {
         type uint8 {
           range "0 .. 31";
         }
         description
           "OSPFv3 instance ID.";
       }
     }

     grouping ospfv3-interface-state {
       description
         "OSPFv3 interface specific operational state.";

       leaf interface-id {
         type uint16;
         config false;
         description
           "OSPFv3 interface ID.";
       }
     }
```

```
     grouping lsa-identifiers {
       description
         "The parameters that uniquely identify an LSA.";
       leaf area-id {
         type area-id-type;
         description
           "Area ID";
       }
       leaf link-id {
         type union {
           type inet:ipv4-address;
           type yang:dotted-quad;
         }
         description "Link ID.";
       }
       leaf type {
         type uint16;
         description
           "LSA type.";
       }
       leaf lsa-id {
         type yang:dotted-quad;
         description "LSA ID.";
       }
       leaf adv-router {
         type yang:dotted-quad;
         description
           "LSA advertising router.";
       }
       leaf seq-num {
         type uint32;
         description
           "LSA sequence number.";
       }
     }

     grouping spf-log {
       description
         "Grouping for SPF log.";
       container spf-log {
         config false;
         description
           "This container lists the SPF log.";
         list event {
           key id;
           description
             "List of SPF logs.
              It is used as a wrapping buffer.";
```

```
          leaf id {
            type uint32;
            description
              "This leaf defines the event identifier.
               This is a purely internal value.";
          }
          leaf spf-type {
            type enumeration {
              enum full {
                description
                  "Computation done is a Full SPF.";
              }
              enum intra {
                description
                  "Computation done is only for intra-area routes.";
              }
              enum inter {
                description
                  "Computation done is only for inter-area
                   summary routes.";
              }
              enum external {
                description
                  "Computation done is only for AS external routes.";
              }
            }
            description
              "The SPF computation type.";
          }
          leaf schedule-timestamp {
            type yang:timestamp;
            description
              "This leaf describes the timestamp
               when the computation was scheduled.";
          }
          leaf start-timestamp {
            type yang:timestamp;
            description
              "This leaf describes the timestamp
               when the computation was started.";
          }
          leaf end-timestamp {
            type yang:timestamp;
            description
              "This leaf describes the timestamp
               when the computation was completed.";
          }
          list trigger-lsa {
```

```
            description
              "The list of LSAs that triggered the computation.";
            uses lsa-identifiers;
          }
        }
      }
    }

    grouping lsa-log {
      description
        "Grouping for LSA log.";
      container lsa-log {
        config false;
        description
          "This conatiner lists the LSA log.
           Local LSA modifications are also included
           in the list.";
        list event {
          key id;
          description
            "List of LSA logs.
             It is used as a wrapping buffer.";
          leaf id {
            type uint32;
            description
              "This leaf defines the event identifier.
               This is a purely internal value.";
          }
          container lsa {
            description
              "This container describes the logged LSA.";
            uses lsa-identifiers;
          }
          leaf received-timestamp {
            type yang:timestamp;
            description
              "This leaf describes the timestamp
               when the LSA was received.  In case of
               local LSA update, the timestamp refers
               to the local LSA update time.";
          }
          leaf reason {
            type identityref {
              base lsa-log-reason;
            }
            description
              "This leaf describes the reason
               that resulted in this LSA log.";
```

```
          }
        }
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol" {
       when "rt:type = 'ospf:ospfv2' or rt:type = 'ospf:ospfv3'" {
        description
          "This augmentation is only valid for a routing protocol
           instance of OSPF (type 'ospfv2' or 'ospfv3').";
      }
      description "OSPF augmentation.";

      container ospf {
        description
          "OSPF.";

        uses ospf-config;
        uses ospf-state;

        list instance {
          key "af";
          description
            "An OSPF routing protocol instance.";

          leaf af {
            type iana-rt-types:address-family;
            description
              "Address-family of the instance.";
          }

          uses instance-config;
          uses instance-state;

          container areas {
            description "All areas.";
            list area {
              key "area-id";
              description
                "List of OSPF areas";
              leaf area-id {
                type area-id-type;
                description
                  "Area ID.";
              }

              uses area-config;
```

```
              uses area-state;

              container virtual-links {
                when "../area-id = '0.0.0.0' and "
                   + "../area-type = 'ospf:normal'" {
                  description
                    "Virtual links must be in backbone area.";
                  }
                description "All virtual links.";
                list virtual-link {
                  key "transit-area-id router-id";
                  description
                    "OSPF virtual link";
                  leaf transit-area-id {
                    type leafref {
                      path "../../../../area/area-id";
                    }
                    must "../../../../area[area-id=current()]/"
                       + "area-id != '0.0.0.0' and "
                       + "../../../../area[area-id=current()]/"
                       + "area-type = 'ospf:normal'" {
                      error-message "Virtual link transit area must "
                                  + "be non-zero.";
                      description
                        "Virtual-link trasit area must be
                         non-zero area.";
                    }
                    description
                      "Virtual link tranist area ID.";
                  }
                  leaf router-id {
                    type rt-types:router-id;
                    description
                      "Virtual Link remote endpoint router ID.";
                  }

                  uses virtual-link-config;
                  uses virtual-link-state;
                }
              }
              container sham-links {
                if-feature pe-ce-protocol;
                description "All sham links.";
                list sham-link {
                  key "local-id remote-id";
                  description
                    "OSPF sham link";
                  leaf local-id {
```

```
                  type inet:ip-address;
                  description
                    "Address of the local Sham Link endpoint.";
                }
                leaf remote-id {
                  type inet:ip-address;
                  description
                    "Address of the remote Sham Link endpoint.";
                }
                uses sham-link-config;
                uses sham-link-state;
              }
            }
            container interfaces {
              description "All interfaces.";
              list interface {
                key "name";
                description
                  "List of OSPF interfaces.";
                leaf name {
                  type if:interface-ref;
                  description
                    "Interface name.";
                }
                uses interface-config;
                uses interface-state;
              } // list of interfaces
            }
          } // list of areas
        }
      } // list of instance
    } // container ospf
  }

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/ospf:ospf/ospf:instance" {
    when "../../rt:type = 'ospf:ospfv2' or
          ../../rt:type = 'ospf:ospfv3'" {
      description
        "This augmentation is only valid for OSPF
        (type 'ospfv2' or 'ospfv3').";
    }
    if-feature multi-topology;
    description
      "OSPF multi-topology instance configuration
       state augmentation.";
    container topologies {
      description "All topologies.";
```

```
          list topology {
            // Topology must be in the same routing-instance
            //  and of same AF as the container.
            key "name";
            description "OSPF topology.";
            leaf name {
              type leafref {
                path "../../../../../../../rt:ribs/rt:rib/rt:name";
              }
              description "RIB";
            }

            uses multi-topology-config;
            uses multi-topology-state;

            container areas {
              description "All areas in the topology.";
              list area {
                key "area-id";
                description
                  "List of OSPF areas";
                leaf area-id {
                  type area-id-type;
                  description
                    "Area ID.";
                }
                uses multi-topology-area-config;
                uses multi-topology-area-state;
              }
            }
          }
        }
      }

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/ospf:ospf/ospf:instance/"
            + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
        when "../../../../../../rt:type = 'ospf:ospfv2'" {
          description
            "This augmentation is only valid for OSPFv2.";
        }
        if-feature ospf:multi-topology;
        description
          "OSPF multi-topology interface configuration state
           augmentation.";
        container topologies {
          description "All topologies for the interface.";
          list topology {
```

```
          key "name";
          description "OSPF interface topology.";
          leaf name {
            type leafref {
              path "../../../../../../../../../../../../"
                 + "rt:ribs/rt:rib/rt:name";
            }
            description
              "One of the topologies enabled on this interface.";
          }

          uses multi-topology-interface-config;
          uses multi-topology-interface-state;
        }
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/ospf:instance/"
          + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
      when "../../../../../../rt:type = 'ospf:ospfv3'" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "OSPFv3 interface specific configuration state
         augmentation.";
      uses ospfv3-interface-config;
      uses ospfv3-interface-state;
    }

    grouping route-content {
      description
        "This grouping defines OSPF-specific route attributes.";
      leaf metric {
        type uint32;
        description "OSPF route metric.";
      }
      leaf tag {
        type uint32;
        default "0";
        description "OSPF route tag.";
      }
      leaf route-type {
        type route-type;
        description "OSPF route type";
      }
    }
```

```
     augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
       when "rt:source-protocol = 'ospf:ospfv2' or "
          + "rt:source-protocol = 'ospf:ospfv3'" {
         description
           "This augmentation is only valid for a routes whose
            source protocol is OSPF.";
       }
       description
         "OSPF-specific route attributes.";
       uses route-content;
     }

     /*
      * RPCs
      */

     rpc clear-neighbor {
       description
         "This RPC request clears a particular
          set of OSPF neighbors. If the operation
          fails for OSPF internal reason, then
          error-tag and error-app-tag should be set
          to a meaningful value.";
       input {
         leaf routing-protocol-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory "true";
           description
             "Name of the OSPF protocol instance which information
              is being queried.

              If the OSPF instance with name equal to the
              value of this parameter doesn't exist, then this
              operation SHALL fail with error-tag 'data-missing'
              and error-app-tag
              'routing-protocol-instance-not-found'.";
         }

         leaf interface {
           type if:interface-ref;
             description
               "Name of the OSPF interface.

                If the OSPF interface with name equal to the
                value of this parameter doesn't exist, then this
```

```
               operation SHALL fail with error-tag 'data-missing'
               and error-app-tag
               'ospf-interface-not-found'.";
        }
      }
    }

    rpc clear-database {
      description
        "This RPC request clears a particular
         OSPF database. If the operation
         fails for OSPF internal reason, then
         error-tag and error-app-tag should be set
         to a meaningful value.";
      input {
        leaf routing-protocol-name {
          type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
               + "rt:control-plane-protocol/rt:name";
          }
          mandatory "true";
          description
            "Name of the OSPF protocol instance whose
             information is being queried.

             If the OSPF instance with name equal to the
             value of this parameter doesn't exist, then this
             operation SHALL fail with error-tag 'data-missing'
             and error-app-tag
             'routing-protocol-instance-not-found'.";
        }
      }
    }

    /*
     * Notifications
     */

    grouping notification-instance-hdr {
      description
       "This group describes common instance specific
        data for notifications.";

      leaf routing-protocol-name {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
             + "rt:control-plane-protocol/rt:name";
        }
```

```
         must "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol[rt:name=current()]/"
            + "rt:type = 'ospf:ospfv2' or "
            + "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol[rt:name=current()]/"
            + "rt:type = 'ospf:ospfv3'";
         description
          "OSPF routing protocol instance name.";
       }

       leaf af {
         type leafref {
           path "/rt:routing/"
              + "rt:control-plane-protocols/rt:control-plane-protocol"
              + "[rt:name=current()/../routing-protocol-name]/"
              + "ospf:ospf/ospf:instance/af";
         }
         description
           "Address family of the OSPF instance.";
       }
     }

     grouping notification-interface {
       description
         "This grouping provides the interface information
          for the notifications.";

       choice if-link-type-selection {
         description
           "Options for link type.";
         container interface {
           description "Normal interface.";
           leaf interface {
             type if:interface-ref;
             description "Interface.";
           }
         }
         container virtual-link {
           description "virtual-link.";
           leaf transit-area-id {
             type area-id-type;
             description "Area ID.";
           }
           leaf neighbor-router-id {
             type rt-types:router-id;
             description "Neighbor Router ID.";
           }
         }
```

```
          container sham-link {
            description "sham-link.";
            leaf area-id {
              type area-id-type;
              description "Area ID.";
            }
            leaf local-ip-addr {
              type inet:ip-address;
              description "Sham link local address.";
            }
            leaf remote-ip-addr {
              type inet:ip-address;
              description "Sham link remote address.";
            }
          }
        }
      }

      grouping notification-neighbor {
        description
          "This grouping provides the neighbor information
           for the notifications.";

        leaf neighbor-router-id {
          type rt-types:router-id;
          description "Neighbor Router ID.";
        }

        leaf neighbor-ip-addr {
          type yang:dotted-quad;
          description "Neighbor address.";
        }
      }

      notification if-state-change {
        uses notification-instance-hdr;
        uses notification-interface;

        leaf state {
          type if-state-type;
          description "Interface state.";
        }
        description
          "This notification is sent when an interface
          state change is detected.";
      }

      notification if-config-error {
```

```
       uses notification-instance-hdr;
       uses notification-interface;

       leaf packet-source {
         type yang:dotted-quad;
         description "Source address.";
       }

       leaf packet-type {
         type packet-type;
         description "OSPF packet type.";
       }

       leaf error {
         type enumeration {
           enum "badVersion" {
             description "Bad version.";
           }
           enum "areaMismatch" {
             description "Area mistmatch.";
           }
           enum "unknownNbmaNbr" {
             description "Unknown NBMA neighbor.";
           }
           enum "unknownVirtualNbr" {
             description "Unknown virtual link neighbor.";
           }
           enum "authTypeMismatch" {
             description "Auth type mismatch.";
           }
           enum "authFailure" {
             description "Auth failure.";
           }
           enum "netMaskMismatch" {
             description "Network mask mismatch.";
           }
           enum "helloIntervalMismatch" {
             description "Hello interval mismatch.";
           }
           enum "deadIntervalMismatch" {
             description "Dead interval mismatch.";
           }
           enum "optionMismatch" {
             description "Option mismatch.";
           }
           enum "mtuMismatch" {
             description "MTU mismatch.";
           }
```

```
          enum "duplicateRouterId" {
            description "Duplicate router ID.";
          }
          enum "noError" {
            description "No error.";
          }
        }
        description "Error code.";
      }
      description
        "This notification is sent when interface
         config error is detected.";
    }

    notification nbr-state-change {
      uses notification-instance-hdr;
      uses notification-interface;
      uses notification-neighbor;

      leaf state {
        type nbr-state-type;
        description "Neighbor state.";
      }

      description
        "This notification is sent when neighbor
         state change is detected.";
    }

    notification nbr-restart-helper-status-change {
      uses notification-instance-hdr;
      uses notification-interface;
      uses notification-neighbor;

      leaf status {
        type restart-helper-status-type;
        description "Restart helper status.";
      }

      leaf age {
        type uint32;
        units seconds;
        description
          "Remaining time in current OSPF graceful restart
           interval, if the router is acting as a restart
           helper for the neighbor.";
      }
```

```
        leaf exit-reason {
          type restart-exit-reason-type;
          description
            "Restart helper exit reason.";
        }
        description
          "This notification is sent when a neighbor restart
           helper status change is detected.";
      }

      notification if-rx-bad-packet {
        uses notification-instance-hdr;
        uses notification-interface;

        leaf packet-source {
          type yang:dotted-quad;
          description "Source address.";
        }

        leaf packet-type {
          type packet-type;
          description "OSPF packet type.";
        }

        description
          "This notification is sent when an OSPF packet that
           cannot be parsed is received on an OSPF interface.";
      }

      notification lsdb-approaching-overflow {
        uses notification-instance-hdr;

        leaf ext-lsdb-limit {
          type uint32;
          description
            "The maximum number of non-default AS-external LSAs
             entries that can be stored in the link state database.";
        }

        description
          "This notification is sent when the number of LSAs
           in the router's link state database has exceeded
           ninety percent of the ext-lsdb-limit.";
      }

      notification lsdb-overflow {
        uses notification-instance-hdr;
```

```
        leaf ext-lsdb-limit {
          type uint32;
          description
            "The maximum number of non-default AS-external LSAs
             entries that can be stored in the link state database.";
        }

        description
          "This notification is sent when the number of LSAs
           in the router’s link state database has exceeded
           ext-lsdb-limit.";
      }

      notification nssa-translator-status-change {
        uses notification-instance-hdr;

        leaf area-id {
          type area-id-type;
          description "Area ID.";
        }

        leaf status {
          type nssa-translator-state-type;
          description
            "NSSA translator status.";
        }

        description
          "This notification is sent when there is a change
           in the router’s role in translating OSPF NSSA LSAs
           to OSPF AS-External LSAs.";
      }

      notification restart-status-change {
        uses notification-instance-hdr;

        leaf status {
          type restart-status-type;
          description
            "Restart status.";
        }

        leaf restart-interval {
          type uint16 {
            range "1..1800";
          }
          units seconds;
          default "120";
```

```
          description
            "Restart interval.";
        }

        leaf exit-reason {
          type restart-exit-reason-type;
          description
            "Restart exit reason.";
        }

        description
          "This notification is sent when the graceful restart
           state for the router has changed.";
      }
    }
    <CODE ENDS>
```

4. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. For OSPF, the ability to modify OSPF configuration will allow the entire OSPF domain to be compromised including peering with unauthorized routers to misroute traffic or mount a massive Denial-of-Service (DoS) attack. The security considerations of OSPFv2 [RFC2328] and [RFC5340].

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The exposure of the Link State Database (LSDB) will expose the detailed topology of the network. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be propritary.

For OSPF authentication, configuration is supported via the specification of key-chains [RFC8177] or the direct specification of key and authentication algorithm. Hence, authentification configuration using the "auth-table-trailer" case in the "authentication" container inherits the security considerations of [RFC8177]. This includes the considerations with respect to the local storage and handling of authentication keys.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The OSPF Yang module support the "clear-neighbor" and "clear-database" RPCs. If access too either of these is compromised, they can result in temporary network outages be employed to mount DoS attacks.

5. Acknowledgements

The authors wish to thank Yi Yang, Alexander Clemm, Gaurav Gupta, Ladislav Lhotka, Stephane Litkowski, Greg Hankins, Manish Gupta and Alan Davey for their thorough reviews and helpful comments.

This document was produced using Marshall Rose's xml2rfc tool.

6. References

6.1. Normative References

[RFC1793] Moy, J., "Extending OSPF to Support Demand Circuits", RFC 1793, DOI 10.17487/RFC1793, April 1995, <https://www.rfc-editor.org/info/rfc1793>.

[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>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <https://www.rfc-editor.org/info/rfc3101>.

[RFC3623] Moy, J., Pillay-Esnault, P., and A. Lindem, "Graceful OSPF Restart", RFC 3623, DOI 10.17487/RFC3623, November 2003, <https://www.rfc-editor.org/info/rfc3623>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC4577] Rosen, E., Psenak, P., and P. Pillay-Esnault, "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4577, DOI 10.17487/RFC4577, June 2006, <https://www.rfc-editor.org/info/rfc4577>.

[RFC4750] Joyal, D., Ed., Galecki, P., Ed., Giacalone, S., Ed., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, DOI 10.17487/RFC4750, December 2006, <https://www.rfc-editor.org/info/rfc4750>.

[RFC5187] Pillay-Esnault, P. and A. Lindem, "OSPFv3 Graceful Restart", RFC 5187, DOI 10.17487/RFC5187, June 2008, <https://www.rfc-editor.org/info/rfc5187>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC5643] Joyal, D., Ed. and V. Manral, Ed., "Management Information Base for OSPFv3", RFC 5643, DOI 10.17487/RFC5643, August 2009, <https://www.rfc-editor.org/info/rfc5643>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <https://www.rfc-editor.org/info/rfc5838>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC6565] Pillay-Esnault, P., Moyer, P., Doyle, J., Ertekin, E., and M. Lundberg, "OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol", RFC 6565, DOI 10.17487/RFC6565, June 2012, <https://www.rfc-editor.org/info/rfc6565>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <https://www.rfc-editor.org/info/rfc7223>.

[RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <https://www.rfc-editor.org/info/rfc8022>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

6.2. Informative References

[I-D.ietf-netmod-revised-datastores] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-02 (work in progress), May 2017.

Appendix A. Contributors’ Addreses

Dean Bogdanovic
Volta Networks, Inc.

EMail: dean@voltanet.io

Kiran Koushik Agrahara Sreenivasa
Cisco Systems
12515 Research Blvd, Bldg 4
Austin, TX 78681
USA

EMail: kkoushik@cisco.com

Authors’ Addresses

Derek Yeung
Arrcus

EMail: derek@arrcus.com

Yingzhen Qu
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: yingzhen.qu@huawei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

EMail: zzhang@juniper.net

Ing-Wher Chen
Jabil

EMail: ing-wher_chen@jabil.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

EMail: acee@cisco.com

Open Shortest Path First IGP   P. Psenak, Ed.
Internet-Draft   K. Talaulikar
Intended status: Standards Track   Cisco Systems, Inc.
Expires: November 24, 2017   W. Henderickx
                                                                      Nokia
                                                         P. Pillay-Esnault
                                                                     Huawei
                                                               May 23, 2017

OSPF LLS Extensions for Local Interface ID Advertisement
draft-ppsenak-ospf-lls-interface-id-01

Abstract

This draft describes the extensions to OSPF link-local signaling to advertise Local Interface Identifier.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Table of Contents

1. Introduction

Every interface is assigned an Interface ID, which uniquely identifies the interface on the router. For example, some implementations MAY be able to use the MIB-II IfIndex [RFC2863] as the Interface ID.

Local/Remote Interface Identifiers MAY be flooded by OSPF [RFC2328] as defined in [RFC4203]. From the perspective of the advertising router, the Local Interface Identifier is a known value, however the Remote Interface Identifier needs to be learnt before it can be advertised. [RFC4203] suggests to use TE Link Local LSA [RFC3630] to communicate Local Interface Identifier to neighbors on the link. Though such mechanism works, it has some drawbacks.

This draft proposes an extension to OSPF link-local signaling (LLS) [RFC5613] to advertise the Local Interface Identifier.

2. Interface ID Exchange using TE Opaque LSA

Usage of the Link Local TE Opaque LSA to propagate the Local Interface Identifier to the neighbors on the link is described in [RFC4203]. This mechanism has following problems:

LSAs can only be flooded over an existing adjacency that is in Exchange state or greater. The adjacency state machine progresses independently on each side of the adjacency and, as such, may reach the Full state on one side before the TE Link Opaque LSA arrives. The consequence is that link can be initially advertised without the Remote Interface Identifier. Later when the TE Link Opaque LSA arrives, the link must be advertised again, this time with the valid Remote Interface Identifier. Implementation may choose to wait before advertising the link, but there is no guarantee that the neighbor will ever advertise the TE Link Opaque LSA with the Interface Identifier. In summary, the existing mechanism does not guarantee that Remote Interface Identifier is known at the time the link is advertised.

TE Opaque LSA is defined for MPLS Traffic Engineering, but the knowledge of the Remote Interface Identifier is useful for other cases where MPLS TE is not used. One example is the lack of valid 2-way connectivity check for remote parallel point-to-point links in OSPF. In such case, TE Opaque LSAs are not exchanged solely for 2-way connectivity correctness.

3. Interface ID Exchange using OSPF LLS

To address the problems described earlier and to allow the Interface Identifiers exchange to be part of the neighbor discovery process, we propose to extend OSPF link-local signaling to advertise the Local Interface Identifier in OSPF Hello packets.

3.1. Local Interface Identifier TLV

The Local Interface Identifier TLV is a new LLS TLV. It has 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Local Interface Identifier                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 18

Length: 4 octet

Local Interface Identifier: The value of the local Interface Identifier.

Local Interface Identifier TLV MUST be present in all Hello packets on all link types, except packets that are sent to the remote end of the virtual-link.

4. Backward Compatibility with RFC 4203

Implementations which support Local Interface ID signalling using LLS MUST prefer the Local Interface ID value received through LLS over the value received through the Link Local TE Opaque LSAs.

Implementations which also support the Local Interface ID signalling via Link Local TE Opaque LSA MAY continue to do so to ensure backward compatibility and they MUST signal the same local interface id via both mechanisms.

During the rare conditions, when the Local Interface ID changes, a timing interval may exist, where the received values of the Local Interface ID advertised through LLS and Link Local TE Opaque LSA may differ. Such situation is temporary and received values via both mechanisms should become equal as soon as the next Hello and/or Link Local TE Opaque LSA is re-generated by the originator.

5. IANA Considerations

This specification updates Link Local Signalling TLV Identifiers registry.

Following values is allocated:

o 18 - Local Interface Identifier TLV

6. Security Considerations

Implementations must assure that malformed LLS TLV and Sub-TLV permutations do not result in errors which cause hard OSPF failures.

7. Contributors

8. Acknowledgements

Thanks to Tony Przygienda for his extensive review and useful comments.

9. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <http://www.rfc-editor.org/info/rfc2863>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <http://www.rfc-editor.org/info/rfc4203>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <http://www.rfc-editor.org/info/rfc5340>.

[RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <http://www.rfc-editor.org/info/rfc5613>.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems, Inc.
Apollo Business Center
Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Ketan Jivan Talaulikar
Cisco Systems, Inc.
S.No. 154/6, Phase I, Hinjawadi
PUNE, MAHARASHTRA   411 057
India

Email: ketant@cisco.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
BE

Email: wim.henderickx@nokia.com

Padma Pillay-Esnault
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: padma@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 25, 2017

P. Psenak
A. Lindem
L. Ginsberg
Cisco Systems
W. Henderickx
Nokia
J. Tantsura
Individual
H. Gredler
RtBrick Inc.
June 23, 2017

OSPFv2 Link Traffic Engineering (TE) Attribute Reuse

draft-ppsenak-ospf-te-link-attr-reuse-05.txt

Abstract

Various link attributes have been defined in OSPFv2 in the context of the MPLS Traffic Engineering (TE) and GMPLS. Many of these link attributes can be used for purposes other than MPLS Traffic Engineering or GMPLS. This documents defines how to distribute such attributes in OSPFv2 for applications other than MPLS Traffic Engineering or GMPLS purposes.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Various link attributes have been defined in OSPFv2 [RFC2328] in the context of the MPLS traffic engineering and GMPLS. All these attributes are distributed by OSPFv2 as sub-TLVs of the Link-TLV advertised in the OSPFv2 TE Opaque LSA [RFC3630].

Many of these link attributes are useful outside of the traditional MPLS Traffic Engineering or GMPLS. This brings its own set of problems, in particular how to distribute these link attributes in OSPFv2 when MPLS TE or GMPLS are not deployed or are deployed in parallel with other applications that use these link attributes.

[RFC7855] discusses use cases/requirements for SR. Included among these use cases is SRTE. If both RSVP-TE and SRTE are deployed in a network, link attribute advertisements can be used by one or both of these applications. As there is no requirement for the link attributes advertised on a given link used by SRTE to be identical to the link attributes advertised on that same link used by RSVP-TE, there is a clear requirement to indicate independently which link attribute advertisements are to be used by each application.

As the number of applications which may wish to utilize link attributes may grow in the future, an additional requirement is that the extensions defined allow the association of additional applications to link attributes without altering the format of the advertisements or introducing new backwards compatibility issues.

Finally, there may still be many cases where a single attribute value can be shared among multiple applications, so the solution should minimize advertising duplicate link/attribute when possible.

1.1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Link attributes examples

This section lists some of the link attributes originally defined for MPLS Traffic Engineering that can be used for other purposes in OSPFv2. The list doesn’t necessarily contain all the required attributes.

1. Remote Interface IP address [RFC3630] - OSPFv2 currently cannot distinguish between parallel links between two OSPFv2 routers. As a result, the two-way connectivity check performed during SPF

may succeed when the two routers disagree on which of the links to use for data traffic.

2. Link Local/Remote Identifiers - [RFC4203] - Used for the two-way connectivity check for parallel unnumbered links. Also used for identifying adjacencies for unnumbered links in Segment Routing traffic engineering.

3. Shared Risk Link Group (SRLG) [RFC4203] - In IPFRR, the SRLG is used to compute diverse backup paths [RFC5714].

4. Unidirectional Link Delay/Loss Metrics [RFC7471] - Could be used for the shortest path first (SPF) computation using alternate metrics within an OSPF area.

3. Advertising Link Attributes

This section outlines possible approaches for advertising link attributes originally defined for MPLS Traffic Engineering purposes or GMPLS when they are used for other applications.

3.1. TE Opaque LSA

One approach for advertising link attributes is to continue to use TE Opaque LSA ([RFC3630]). There are several problems with this approach:

1. Whenever the link is advertised in a TE Opaque LSA, the link becomes a part of the TE topology, which may not match IP routed topology. By making the link part of the TE topology, remote nodes may mistakenly believe that the link is available for MPLS TE or GMPLS, when, in fact, MPLS is not enabled on the link.

2. The TE Opaque LSA carries link attributes that are not used or required by MPLS TE or GMPLS. There is no mechanism in a TE Opaque LSA to indicate which of the link attributes are passed to MPLS TE application and which are used by other applications including OSPFv2 itself.

3. Link attributes used for non-TE purposes are partitioned across multiple LSAs - the TE Opaque LSA and the Extended Link Opaque LSA. This partitioning will require implementations to lookup multiple LSAs to extract link attributes for a single link, bringing needless complexity to OSPFv2 implementations.

The advantage of this approach is that there is no additional standardization requirement to advertise the TE/GMPL attributes for other applications. Additionally, link attributes are only advertised once when both OSPF TE and other applications are deployed on the same link. This is not expected to be a common deployment scenario.

3.2. Extended Link Opaque LSA

An alternative approach for advertising link attributes is to use Extended Link Opaque LSAs as defined in [RFC7684]. This LSA was defined as a generic container for distribution of the extended link attributes. There are several advantages in using Extended Link LSA:

1. Advertisement of the link attributes does not make the link part of the TE topology. It avoids any conflicts and is fully compatible with the [RFC3630].

2. The TE Opaque LSA remains truly opaque to OSPFv2 as originally defined in [RFC3630]. Its content is not inspected by OSPFv2 and OSPFv2 acts as a pure transport.

3. There is clear distinction between link attributes used by TE and link attributes used by other OSPFv2 applications.

4. All link attributes that are used by OSPFv2 applications are advertised in a single LSA, the Extended Link Opaque LSA.

The disadvantage of this approach is that in rare cases, the same link attribute is advertised in both the TE Opaque and Extended Link Attribute LSAs. Additionally, there will be additional standardization effort. However, this could also be viewed as an advantage as the non-TE use cases for the TE link attributes are documented and validated by the OSPF working group.

3.3. Selected Approach

It is RECOMMENDED to use the Extended Link Opaque LSA ([RFC7684] to advertise any link attributes used for non-TE purposes in OSPFv2, including those that have been originally defined for TE purposes. TE link attributes used for TE purposes continue to use TE Opaque LSA ([RFC3630]).

It is also RECOMMENDED to keep the format of the link attribute TLVs that have been defined for TE purposes unchanged even when they are used for non-TE purposes.

Finally, it is RECOMMENDED to allocate unique code points for link attribute TLVs that have been defined for TE purposes for the OSPFv2 Extended Link TLV Sub-TLV Registry as defined in [RFC7684]. For each reused TLV, the code point will be defined in an IETF document along with the expected usecase(s).

4. Reused TE link attributes

This section defines the use case and code points for the OSPFv2 Extended Link TLV Sub-TLV Registry for some of the link attributes that have been originally defined for TE or GMPLS purposes.

4.1. Remote interface IP address

The OSPFv2 description of an IP numbered point-to-point adjacency does not include the remote IP address. As described in Section 2, this makes the two-way connectivity check ambiguous in the presence of the parallel point-to-point links between two OSPFv2 routers.

The Remote IP address of the link can also be used for Segment Routing traffic engineering to identify the link in a set of parallel links between two OSPFv2 routers [I-D.ietf-ospf-segment-routing-extensions]. Similarly, the remote IP address is useful in identifying individual parallel OSPF links advertised in BGP Link-State as described in [I-D.ietf-idr-ls-distribution].

To advertise the Remote interface IP address in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 2.5.4. of [RFC3630] is used and TLV type TBD1 is used.

4.2. Link Local/Remote Identifiers

The OSPFv2 description of an IP unnumbered point-to-point adjacency does not include the remote link identifier. As described in Section 2, this makes the two-way connectivity check ambiguous in the presence of the parallel point-to-point IP unnumbered links between two OSPFv2 routers.

The local and remote link identifiers can also be used for Segment Routing traffic engineering to identify the link in a set of parallel IP unnumbered links between two OSPFv2 routers [I-D.ietf-ospf-segment-routing-extensions]. Similarly, these identifiers are useful in identifying individual parallel OSPF links advertised in BGP Link-State as described in [I-D.ietf-idr-ls-distribution].

To advertise the link Local/Remote identifiers in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 1.1. of [RFC4203] is used and TLV type TBD2 is used.

4.3. Shared Risk Link Group (SRLG)

The SRLG of a link can be used in IPFRR to compute a backup path that does not share any SRLG group with the protected link.

To advertise the SRLG of the link in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 1.3. of [RFC4203] is used and TLV type TBD3 is used.

4.4. Extended Metrics

[RFC3630] defines several link bandwidth types. [RFC7471] defines extended link metrics that are based on link bandwidth, delay and loss characteristics. All these can be used to compute best paths within an OSPF area to satisfy requirements for bandwidth, delay (nominal or worst case) or loss.

To advertise extended link metrics in the OSPFv2 Extended Link TLV, the same format of the sub-TLVs as defined in [RFC7471] is used with following TLV types:

TBD4 - Unidirectional Link Delay

TBD5 - Min/Max Unidirectional Link Delay

TBD6 - Unidirectional Delay Variation

TBD7 - Unidirectional Link Loss

TBD8 - Unidirectional Residual Bandwidth

TBD9 - Unidirectional Available Bandwidth

TBD10 - Unidirectional Utilized Bandwidth

5. Advertisement of Application Specific Values

Multiple applications can utilize link attributes that are flooded by OSPFv2. Some examples of applications using the link attributes are Segment Routing Traffic Engineering and LFA [RFC5286].

In some cases the link attribute only has a single value that is applicable to all applications. An example is a Remote interface IP address [Section 4.1] or Link Local/Remote Identifiers [Section 4.2].

In some cases the link attribute MAY have different values for different applications. An example could be SRLG [Section 4.3], where values used by LFA could be different then the values used by Segment Routing Traffic Engineering.

To allow advertisement of the application specific values of the link attribute, a new Extended Link Attribute sub-TLV of the Extended Link TLV [RFC7471] is defined. The Extended Link Attribute sub-TLV is an optional sub-TLV and can appear multiple times in the Extended Link TLV. It has 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     SABML     |     UDABML    |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Standard Application Bit-Mask                  |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                User Defined Application Bit-Mask              |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link Attribute sub-sub-TLVs              |
+-                                                             -+
|                            ...                                |
```

where:

Type: TBD11, suggested value 14

Length: variable

SABML: Standard Application Bit-Mask Length. If the Standard Application Bit-Mask is not present, the Standard Application Bit-Mask Length MUST be set to 0.

UDABML: User Defined Application Bit-Mask Length. If the User Defined Application Bit-Mask is not present, the User Defined Application Bit-Mask Length MUST be set to 0.

Standard Application Bit-Mask: Optional set of bits, where each bit represents a single standard application. The following bits are defined by this document:

Bit-0: RSVP Traffic Engineering

Bit-1: Segment Routing Traffic Engineering

Bit-2: Loop Free Alternate (LFA). Includes all LFA types.

User Defined Application Bit-Mask: Optional set of bits, where each bit represents a single user defined application.

Standard Application Bits are defined/sent starting with Bit 0. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of octets that will need to be transmitted.

User Defined Application bits have no relationship to Standard Application bits and are NOT managed by IANA or any other standards body. It is recommended that bits are used starting with Bit 0 so as to minimize the number of octets required to advertise all of them.

Undefined bits in both Bit-Masks MUST be transmitted as 0 and MUST be ignored on receipt. Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

If the link attribute advertisement is limited to be used by a specific set of applications, corresponding Bit-Masks MUST be present and application specific bit(s) MUST be set for all applications that use the link attributes advertised in the Extended Link Attribute sub-TLV.

Application Bit-Masks apply to all link attributes that support application specific values and are advertised in the Extended Link Attribute sub-TLV.

The advantage of not making the Application Bit-Masks part of the attribute advertisement itself is that we can keep the format of the link attributes that have been defined previously and reuse the same format when advertising them in the Extended Link Attribute sub-TLV.

If the link attribute is advertised and there is no Application Bit-Mask present in the Extended Link Attribute Sub-TLV, the link attribute advertisement MAY be used by any application. If, however, another advertisement of the same link attribute includes any Application Bit-Mask in the Extended Link Attribute sub-TLV, applications that are listed in the Application Bit-Masks of such Extended Link Attribute sub-TLV SHOULD use the attribute advertisement which has the application specific bit set in the Application Bit-Masks.

If the same application is listed in the Application Bit-Masks of more then one Extended Link Attribute sub-TLV, the application SHOULD use the first advertisement and ignore any subsequent advertisements of the same attribute. This situation SHOULD be logged as an error.

This document defines the set of link attributes for which the Application Bit-Masks may be advertised. If any of the Application Bit-Masks is included in the Extended Link Attribute sub-TLV that advertises any link attribute(s) NOT listed below, the Application Bit-Masks MUST NOT be used for such link attribute(s). It MUST be used for those attribute(s) that support application specific values. Documents which define new link attributes MUST state whether the new attributes support application specific values. The link attributes to which the Application Bit-Masks may apply are:

- Shared Risk Link Group

- Unidirectional Link Delay

- Min/Max Unidirectional Link Delay

- Unidirectional Delay Variation

- Unidirectional Link Loss

- Unidirectional Residual Bandwidth

- Unidirectional Available Bandwidth

- Unidirectional Utilized Bandwidth

6. Deployment Considerations

If link attributes are advertised associated with zero length application bit masks for both standard applications and user defined applications, then that set of link attributes MAY be used by any application. If support for a new application is introduced on any node in a network in the presence of such advertisements, these advertisements MAY be used by the new application. If this is not what is intended, then existing advertisements MUST be readvertised with an explicit set of applications specified before a new application is introduced.

7. Attribute Advertisements and Enablement

This document defines extensions to support the advertisement of application specific link attributes. The presence or absence of link attribute advertisements for a given application on a link does NOT indicate the state of enablement of that application on that link. Enablement of an application on a link is controlled by other means.

For some applications, the concept of enablement is implicit. For example, SRTE implicitly is enabled on all links which are part of the Segment Routing enabled topology. Advertisement of link attributes supports constraints which may be applied when specifying an explicit path through that topology.

For other applications enablement is controlled by local configuration. For example, use of a link as an LFA can be controlled by local enablement/disablement and/or the use of administrative tags.

It is an application specific policy as to whether a given link can be used by that application even in the absence of any application specific link attributes.

8. Backward Compatibility

Link attributes may be concurrently advertised in both the TE Opaque LSA [RFC3630] and the Extended Link Opaque LSA [RFC7684].

In fact, there is at least one OSPF implementation that utilizes the link attributes advertised in TE Opaque LSAs [RFC3630] for Non-RSVP TE applications. For example, this implementation of LFA and remote LFA utilizes links attributes such as Shared Risk Link Groups (SRLG) [RFC4203] and Admin Group [[RFC3630]advertised in TE Opaque LSAs. These applications are described in [RFC5286], [RFC7490], [I-D.ietf-rtgwg-lfa-manageability] and [I-D.psarkar-rtgwg-rlfa-node-protection].

When an OSPF routing domain includes routers using link attributes from TE Opaque LSAs for Non-RSVP TE applications such as LFA, OSPF routers in that domain should continue to advertise such TE Opaque LSAs. If there are also OSPF routers using the link attributes described herein for any application, OSPF routers in the routing domain will also need to advertise these attributes in OSPF Extended Link Attributes LSAs [RFC7684]. In such a deployment, the advertised attributes SHOULD be the same and Non-RSVP application access to link attributes is a matter of local policy.

9. Security Considerations

Implementations must assure that malformed TLV and Sub-TLV permutations do not result in errors that cause hard OSPFv2 failures.

10. IANA Considerations

OSPFv2 Extended Link TLV Sub-TLVs registry [RFC7684] defines sub-TLVs at any level of nesting for OSPFv2 Extended Link TLVs. This specification updates OSPFv2 Extended Link TLV sub-TLVs registry with the following TLV types:

TBD1 (4 Recommended) - Remote interface IP address

TBD2 (5 Recommended) - Link Local/Remote Identifiers

TBD3 (6 Recommended) - Shared Risk Link Group

TBD4 (7 Recommended) - Unidirectional Link Delay

TBD5 (8 Recommended) - Min/Max Unidirectional Link Delay

TBD6 (9 Recommended) - Unidirectional Delay Variation

TBD7 (10 Recommended) - Unidirectional Link Loss

TBD8 (11 Recommended) - Unidirectional Residual Bandwidth

TBD9 (12 Recommended) - Unidirectional Available Bandwidth

TBD10 (13 Recommended) - Unidirectional Utilized Bandwidth

TBD11 (14 Recommended) - Extended Link Attribute

This specification defines a new Link-Attribute-Applicability Application Bits registry and defines following bits:

Bit-0 - Segment Routing Traffic Engineering

Bit-1 - LFA

11. Acknowledgments

Thanks to Chris Bowers for his review and comments.

12. References

12.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <http://www.rfc-editor.org/info/rfc5714>.

[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, <http://www.rfc-editor.org/info/rfc7684>.

12.2. Informative References

[I-D.ietf-idr-ls-distribution] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-13 (work in progress), October 2015.

[I-D.ietf-ospf-segment-routing-extensions] Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-16 (work in progress), May 2017.

[I-D.ietf-rtgwg-lfa-manageability] Litkowski, S., Decraene, B., Filsfils, C., Raza, K., and M. Horneffer, "Operational management of Loop Free Alternates", draft-ietf-rtgwg-lfa-manageability-11 (work in progress), June 2015.

[I-D.psarkar-rtgwg-rlfa-node-protection] psarkar@juniper.net, p., Gredler, H., Hegde, S., Bowers, C., Litkowski, S., and H. Raghuveer, "Remote-LFA Node Protection and Manageability", draft-psarkar-rtgwg-rlfa-node-protection-05 (work in progress), June 2014.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <http://www.rfc-editor.org/info/rfc4203>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <http://www.rfc-editor.org/info/rfc5286>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <http://www.rfc-editor.org/info/rfc7471>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <http://www.rfc-editor.org/info/rfc7490>.

[RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements", RFC 7855, DOI 10.17487/RFC7855, May 2016, <http://www.rfc-editor.org/info/rfc7855>.

Authors' Addresses

Peter Psenak
Cisco Systems
Apollo Business Center
Mlynske nivy 43
Bratislava, 821 09
Slovakia

Email: ppsenak@cisco.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

Email: acee@cisco.com

Les Ginsberg
Cisco Systems
821 Alder Drive
MILPITAS, CA 95035
USA

Email: ginsberg@cisco.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp, 2018 94089
Belgium

Email: wim.henderickx@nokia.com

Jeff Tantsura
Individual
USA

Email: jefftant.ietf@gmail.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

X. Xu
Huawei
L. Fang
ebay
J. Tantsura
Individual

October 30, 2017

OSPF Flooding Reduction in MSDC
draft-xu-ospf-flooding-reduction-in-msdc-02

Abstract

OSPF is commonly used as an underlay routing protocol for MSDC (Massively Scalable Data Center) networks. For a given OSPF router within the CLOS topology, it would receive multiple copies of exactly the same LSA from multiple OSPF neighbors. In addition, two OSPF neighbors may send each other the same LSA simultaneously. The unneccessary link-state information flooding wastes the precious process resource of OSPF routers greatly due to the fact that there are too many OSPF neighbors for each OSPF router within the CLOS topology. This document proposes some extensions to OSPF so as to reduce the OSPF flooding within MSDC networks greatly. The reduction of the OSPF flooding is much beneficial to improve the scalability of MSDC networks. These modifications are applicable to both OSPFv2 and OSPFv3.

Copyright Notice

Table of Contents

1. Introduction

OSPF is commonly used as an underlay routing protocol for Massively Scalable Data Center (MSDC) networks where CLOS is the most popular toplogy. For a given OSPF router within the CLOS topology, it would receive multiple copies of exactly the same LSA from multiple OSPF neighbors. In addition, two OSPF neighbors may send each other the same LSA simultaneously. The unnecessary link-state information flooding wastes the precious process resource of OSPF routers greatly and therefore OSPF could not scale very well in MSDC networks.

To simplify the network management task, centralized controllers are becoming fundamental network elements in most MSDCs. One or more controllers are usually connected to all routers within the MSDC network via a Local Area Network (LAN) which is dedicated for network management purpose (called management LAN), as shown in Figure 1.

```
           +----------+                  +----------+
           |Controller|                  |Controller|
           +----+-----+                  +-----+----+
                |DR                            |BDR
                |                              |
                |                              |
   ---+---------+---+----------+-----------+---+---------+-Management LAN
      |             |          |           |             |
      |Non-DR       |Non-DR    |Non-DR     |Non-DR       |Non-DR
      |             |          |           |             |
      |         +---+--+       |       +---+--+          |
      |         |Router|       |       |Router|          |
      |         *------*-      |      /*---/--*          |
      |        /     \   --    |    //    /    \         |
      |        /     \     --  |  //      /    \         |
      |       /       \      --|//       /      \        |
      |       /        \      /*-       /        \       |
      |      /          \   // | --    /         \       |
      |      /          \ //   |   --  /          \      |
      |     /           /X     |     --           \      |
      |     /         //  \    |     / --          \     |
      |    /        //    \    |     /   --         \    |
      |    /      //       \   |    /      --       \    |
      |   /     //          \  |   /         --      \   |
      |   /   //             \ |  /            --     \  |
      |  /  //               \ |  /              --   \  |
    +-+- //*                +\\+-/-+               +---\-++
    |Router|                |Router|               |Router|
    +------+                +------+               +------+
```

Figure 1

With the assistance of controllers acting as OSPF Designated Router (DR)/Backup Designated Router (BDR) for the management LAN, OSPF routers within the MSDC network don't need to exchange any other types of OSPF packet than the OSPF Hello packet among them. As specified in [RFC2328], these Hello packets are used for the purpose of establishing and maintaining neighbor relationships and ensuring bidirectional communication between OSPF neighbors, and even the DR/BDR election purpose in the case where those OSPF routers are connected to a broadcast network. In order to obtain the full topology information (i.e., the fully synchronized link-state database) of the MSDC's network, these OSPF routers just need to exchange the link-state information with the controllers being elected as OSPF DR/BDR for the management LAN instead.

To further suppress the flooding of multicast OSPF packets originated from OSPF routers over the management LAN, OSPF routers would not send multicast OSPF Hello packets over the management LAN. Insteads, they just wait for OSPF Hello packets originated from the controllers being elected as OSPF DR/BDR initially. Once OSPF DR/BDR for the management LAN have been discovered, they start to send OSPF Hello packets directly (as unicasts) to OSPF DR/BDR periodically. In addition, OSPF routers would send other types of OSPF packets (e.g., Database Descriptor packet, Link State Request packet, Link State Update packet, Link State Acknowledgment packet) to OSPF DR/BDR for the management LAN as unicasts as well. In contrast, the controllers being elected as OSPF DR/BDR would send OSPF packets as specified in [RFC2328]. As a result, OSPF routers would not receive OSPF packets from one another unless these OSPF packets are forwarded as unknown unicasts over the management LAN. Through the above modifications to the current OSPF router behaviors, the OSPF flooding is greatly reduced, which is much beneficial to improve the scalability of MSDC networks. These modifications are applicable to both OSPFv2 [RFC2328] and OSPFv3 [RFC5340].

Furthermore, the mechanism for OSPF refresh and flooding reduction in stable topologies as described in [RFC4136] could be considered as well.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

This memo makes use of the terms defined in [RFC2328].

3. Modifications to Current OSPF Behaviors

3.1. OSPF Routers as Non-DRs

After the exchange of OSPF Hello packets among OSPF routers, the OSPF neighbor relationship among them would transition to and remain in the TWO-WAY state. OSPF routers would originate Router-LSAs and/or Network-LSAs accordingly depending upon the link-types. Note that the neighbors in the TWO-WAY state would be advertised in the Router-LSAs and/or Network-LSA. This is a little bit different from the OSPF router behavior as specified in [RFC2328] where the neighbors in the TWO-WAY state would not be advertised. However, these self-originated LSAs need not to be exchanged directly among them anymore. Instead, these LSAs just need to be sent solely to the controllers being elected as OSPF DR/BDR for the management LAN.

To further reduce the flood of multicast OSPF packets over the management LAN, OSPF routers SHOULD send OSPF packets as unicasts. More specifically, OSPF routers SHOULD send unicast OSPF Hello packets periodically to the controllers being elected as OSPF DR/BDR. In other words, OSPF routers would not send any OSPF Hello packet over the management LAN until they have found OSPF DR/BDR for the management LAN. Note that OSPF routers SHOULD NOT be elected as OSPF DR/BDR for the management LAN (This is done by setting the Router Priority of those OSPF routers to zero). As a result, OSPF routers would not see each other over the management LAN. Furthermore, OSPF routers SHOULD send all other types of OSPF packets than OSPF Hello packets (i.e., Database Descriptor packet, Link State Request packet, Link State Update packet, Link State Acknowledgment packet) to the controllers being elected as OSPF DR/BDR as unicasts as well.

To avoid the data traffic from being forwarded across the management LAN, the cost of all OSPF routers' interfaces to the management LAN SHOULD be set to the maximum value.

When a given OSPF router lost its connection to the management LAN, it SHOULD actively establish FULL adjacency with all of its OSPF neighbors within the CLOS network. As such, it could obtain the full LSDB of the CLOS network while flooding its self-originated LSAs to the remaining part of the whole network. That's to say, for a given OSPF router within the CLOS network, it would not actively establish FULL adjacency with its OSPF neighbor in the TWO-WAY state by default. However, it SHOULD NOT refuse to establish FULL adjacency with a given OSPF neighbors when receiving Database Description Packets from that OSPF neighbor.

3.2. Controllers as DR/BDR

The controllers being elected as OSPF DR/BDR would send OSPF packets as multicasts or unicasts as per [RFC2328]. In addition, Link State Acknowledgment packets are RECOMMENDED to be sent as unicasts rather than multicasts if possible.

4. Acknowledgements

The authors would like to thank Acee Lindem for his valuable comments and suggestions on this document.

5. IANA Considerations

TBD.

6. Security Considerations

TBD.

7. References

7.1. 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>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

7.2. Informative References

[RFC4136] Pillay-Esnault, P., "OSPF Refresh and Flooding Reduction in Stable Topologies", RFC 4136, DOI 10.17487/RFC4136, July 2005, <https://www.rfc-editor.org/info/rfc4136>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <https://www.rfc-editor.org/info/rfc5838>.

Authors’ Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Luyuan Fang
ebay

Email: lufang@ebay.com

Jeff Tantsura
Individual

Email: jefftant@gmail.com