Abstract

This document specifies extensions to OSPF for distributing Traffic Engineering (TE) information on a link in a sequence of time intervals.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 4, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction .............................................. 3
2. Terminology ............................................. 3
3. Conventions Used in This Document ....................... 3
4. Representation of TE Information ......................... 4
   4.1. TE Information in Absolute Time .................... 4
   4.2. TE Information in Relative Time .................... 5
5. Extensions to OSPF ....................................... 6
   5.1. TE LSA ............................................ 6
   5.2. TTS Link TLV .................................... 7
6. Security Considerations ................................ 10
7. IANA Considerations .................................... 10
8. Acknowledgement ......................................... 10
9. References .............................................. 10
   9.1. Normative References ............................... 10
   9.2. Informative References ............................... 10
Authors’ Addresses .................................. 11
1. Introduction

Once an existing multiprotocol label switching (MPLS) traffic engineering (TE) label switched path (LSP) is set up, it is assumed to carry traffic forever until it is down. When an MPLS TE LSP tunnel is up, it is assumed that the LSP consumes its reserved network resources forever even though the LSP may only use network resources during some period of time. As a result, the network resources are not used efficiently. Moreover, a tunnel service can not be reserved or booked in advance for a period of time.

This document specifies extensions to OSPF for supporting the setup of an MPLS TE LSP in a period of time called a time interval or a sequence of time intervals. It is assumed that the LSP carries traffic during this time interval or each of these time intervals. Thus the network resources are efficiently used. More importantly, some new services can be provided. For example, a consumer can book a tunnel service in advance for a time interval or a sequence of time intervals. Tunnel services may be scheduled.

2. Terminology

A Time Interval: a time period from time Ta to time Tb.

LSP: Label Switched Path. An LSP is a P2P (point-to-point) LSP or a P2MP (point-to-multipoint) LSP.

LSP in a time interval: LSP that carries traffic in the time interval.

LSP in a sequence of time intervals: LSP that carries traffic in each of the time intervals.

Temporal LSP: LSP in a time interval or LSP in a sequence of time intervals.

TEDB: Traffic Engineering Database.

This document uses terminologies defined in RFC2328 and RFC3630.

3. 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.
4. Representation of TE Information

The existing Open Shortest Path First (OSPF) Traffic Engineering (TE) distributes an unreserved bandwidth Bi at each of eight priority levels for a link at one point of time, for example, at the current time.

\[
\text{Bandwidth} \uparrow
\]

\[
\begin{array}{c}
\text{Bi} \\
\hline
\end{array}
\]

\[
\rightarrow \text{Timelapse}
\]

This means that the link has bandwidth Bi at a priority level from now to forever until there is a change to it. This TE information on the link is stored in TEDB.

Thus, a temporal LSP (i.e., an LSP in a time interval) cannot be set up using the information in the TEDB and the bandwidth cannot be reserved in advance for the LSP in the time interval.

To support temporal LSPs, we should extend OSPF to distribute TE information on a link in a series of time intervals.

4.1. TE Information in Absolute Time

Suppose that the amount of the unreserved bandwidth at a priority level on a link is Bj in a time interval from time Tj to Tk (k = j+1), where j = 0, 1, 2, .... The unreserved bandwidth on the link can be represented as

\[
[T0, B0], [T1, B1], [T2, B2], [T3, B3], ...
\]

This is an absolute time representation of bandwidths on a link. Time Tj (j = 0, 1, 2, ...) MUST be a synchronized time among all network nodes.
If an LSP is deleted or down at time T and uses bandwidth B, then for every time interval/period (after time T) during which bandwidth B is reserved for the LSP on a link attached to a network node, the network node adds B to the link for that interval/period.

If an LSP is set up at time T and uses bandwidth B, then for every time interval/period (after time T) during which bandwidth B is reserved for the LSP on a link attached to a network node, the network node subtracts bandwidth B from the link for that interval/period.

If there are significant changes to the bandwidths on a link attached to a network node, the network node distributes the bandwidths on the link to other network nodes. That is that a updated [T0, B0], [T1, B1], [T2, B2], [T3, B3], etc., are distributed to other network nodes in the network. Each of the other network nodes can construct or determine the bandwidth for a series of time intervals/periods on a link after receiving the information.

4.2. TE Information in Relative Time

Alternatively, a relative time representation of bandwidths on a link can be used. For example, the amount of the unreserved bandwidth at a priority level on a link is Bj during a series of time intervals/periods can be expressed as

\[[P0, B0], [P1, B1], [P2, B2], [P3, B3], \ldots,\text{ where } P_j = T_k - T_j, k = (j+1) \text{ and } j = 0, 1, 2, 3, \ldots.\]

In this representation, every time T_j (j = 0, 1, 2, ...) can be a local time. A timer may expire after every unit of time (e.g., every second) and trigger --P0, which decrements P0. When P0 = 0, P1 becomes P0, P2 becomes p1, and so on.
If there are significant changes to the bandwidths on a link attached to a node, the node distributes the bandwidths on the link to other nodes. That is that a updated \([P_0, B_0], [P_1, B_1], [P_2, B_2], [P_3, B_3], \ldots\), are distributed to other network nodes in the network. On each of the other network nodes, a timer may expire for every unit of time (e.g., every second) and trigger \(--P_0\), which decrements \(P_0\). When \(P_0 = 0\), \(P_1\) becomes \(P_0\), \(P_2\) becomes \(p_1\), and so on.

An advantage of using relative time representation is that the times or clocks on all the network nodes can be different.

5. Extensions to OSPF

This section describes the extensions to OSPF for supporting the setup of temporal LSPs.

5.1. TE LSA

An opaque LSA of type 10 is originated by a network node to distribute TE information such as the bandwidth of a link that is attached to the network node.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |     Options   |  LS Type=10   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       1       |                 Opaque ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
˜                             TLVs                              ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The opaque LSA comprises a link-state (LS) age field, an options field, an LS type field, an opaque identifier (ID) field, an advertising router field, an LS sequence number field, an LS checksum field, a length field, and one or more TLVs.

The LS age field indicates the time since the LSA was originated in seconds. The options field indicates the optional capabilities.
supported by the described portion of the routing domain. The LS type field indicates the type of the LSA. The opaque ID field is a number used to maintain multiple opaque LSAs. The advertising router field indicates the Router ID of the router that originated the LSA. The LS sequence number field is used to detect old or duplicate LSAs. Successive instances of an LSA are given successive LS sequence numbers. The LS checksum field indicates the Fletcher checksum of the complete contents of the LSA, including the LSA header but excluding the LS age field. The length field indicates the length of the LSA in bytes.

5.2. TTS Link TLV

In addition to existing router address TLV and link TLV, TLVs fields may comprise a new temporal tunnel service (TTS) link TLV.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type (5)           |      Length (variable)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved (0)         |        Segment-Number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          sub TLVs                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The TTS link TLV comprises a type field, a length field, a reserved field, a segment-number field, and a sub TLVs field.

The type field may comprise a value assigned by the Internet Assigned Number Authority (IANA) to indicate that the TLV is a TTS link TLV. For example, the type field may comprise a value of 5. The length field may indicate the length of the values in the TTS link TLV in bytes.

The segment-number indicates a segment of the TTS link TLV. The information on a link may be too big to fit into one TTS link TLV. In this case, the information may be split into a few of segments, each of which is put into a TTS link TLV and identified by a segment number.

The sub TLV field comprises a link type sub-TLV and a link ID sub-TLV. It may further comprise a local address sub-TLV, a remote address sub-TLV, a TE metric sub-TLV, a maximum bandwidth sub-TLV, a maximum reservable bandwidth sub-TLV, an unreserved bandwidth sub-TLV, an administrator group sub-TLV, a relative TTS unreserved
bandwidth sub-TLV, an absolute TTS unreserved bandwidth sub-TLV, and any other suitable sub-TLVs.

The format of an absolute TTS unreserved bandwidth sub-TLV is shown below.

```
+-----------------+------------------+
|                |                  |
| 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               | 3                 |
| +---------------+------------------+
| |                  |
| | Type (21)       |
| |                  |
| +-----------------+------------------+
| |                  |
| | Length (36+36*n)|                  |
| |                  |
| +-----------------+------------------+
| |                  |
| | T0               |
| |                  |
| +-----------------+------------------+
| |                  |
| | B0[8]           |
| |                  |
| +-----------------+------------------+
| |                  |
| | T1               |
| |                  |
| +-----------------+------------------+
| |                  |
| | B1[8]           |
| |                  |
| +-----------------+------------------+
| |                  |
| | Tn               |
| |                  |
| +-----------------+------------------+
| |                  |
| | Bn[8]           |
| |                  |
| +-----------------+------------------+
```

It comprises a type field, a length field, absolute time fields, and unreserved bandwidth fields.

The type field may comprise a value assigned by the IANA to indicate that the sub-TLV is an absolute TTS unreserved bandwidth sub-TLV. For example, the type field may comprise a value of 21.

The length field may indicate the length of the values in the absolute TTS unreserved bandwidth sub-TLV in bytes.

The absolute time fields and the unreserved bandwidth fields may be in pairs such as

```
[ T0, B0[8] ], [ T1, B1[8] ], ..., [ Tn, Bn[8] ],
```
where $T_0$, $T_1$, ..., $T_n$ are the times synchronized among all the nodes and $B_j[8]$ ($j=0, 1, ..., n$) are the amount of unreserved bandwidth at eight priority levels in the time interval/period from $T_j$ to $T_k$ ($k=j+1$).

The format of a relative TTS unreserved bandwidth sub-TLV is illustrated as follows.

```
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 (22)          |      Length (36+36*n)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              P0                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             B0[8]                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              P1                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             B1[8]                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Pn                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Bn[8]                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

It comprises a type field, a length field, relative time fields, and unreserved bandwidth fields.

The type field may comprise a value assigned by the IANA to indicate that the sub-TLV is a relative TTS unreserved bandwidth sub-TLV. For example, the type field may comprise a value of 22.

The length field may indicate the length of the values in the relative TTS unreserved bandwidth sub-TLV in bytes.

The relative time fields and the unreserved bandwidth fields may be in pairs such as

```
[ P0, B0[8] ], [ P1, B1[8] ], ..., [ Pn, Bn[8] ],
```
where $P_j$ ($j=0, 1, \ldots, n$) is the time period during which the unreserved bandwidth is $B_8$, containing the amount of unreserved bandwidth at eight priority levels.

6. Security Considerations

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

7. IANA Considerations

This section specifies requests for IANA allocation.

8. Acknowledgement

The author would like to thank people for their valuable comments on this draft.

9. References

9.1. Normative References


9.2. Informative References

Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001,

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
US
Email: huaimo.chen@futurewei.com

Mehmet Toy
Verizon
USA
Email: mehmet.toy@verizon.com

Vic Liu
China Mobile
No.32 Xuanwumen West Street, Xicheng District
Beijing, 100053
China
Email: liu.cmri@gmail.com

Lei Liu
Fujitsu
USA
Email: lliu@us.fujitsu.com
Abstract

This document proposes a TLV within the body of the OSPF Router Information (RI) Opaque LSA, called Self-defined Sub-TLV Container TLV. Here the term OSPF means both OSPFv2 and OSPFv3. This attribute is meant to accommodate policy-based and deployment-specific use cases.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

There are some use cases where OSPF is used for service auto-discovery by using node administrative tags [I-D.ietf-ospf-node-admin-tag]. One major benefit of using administrative tags rather than IANA defined TLVs or sub-TLVs to indicate different services is to facilitate the rapid deployment of new services without any need for the standardization of those TLVs or sub-TLVs. However, there are some special use cases where the service to be advertised has one or more attributes which need to be advertised as well. In such case, the administrative tag is not much applicable anymore.

To inherit the benefit of administrative tags (i.e., allowing operators to use OSPF for service auto-discovery without the need of any standardization process) while meeting the requirement of advertising services and their associated attributes, this document proposes a TLV within the body of the OSPF Router Information (RI) Opaque LSA, called Self-defined Sub-TLV Container TLV. With such TLV, operators could flexibly define one or more sub-TLVs indicating one or more services and their associated attributes without relying on any standardization process.

The characterization of the TLV and its associated sub-SLVs is local to the each administrative domain. Defining new sub-TLVs is therefore deployment-specific and policy-based. OSPF denotes both OSPFv2 and OSPFv3.
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. Sample Use Cases

There can be several possible use cases and applications for Self-defined Sub-TLV Container TLV defined in Section 4. This section provides few examples how operators can deploy services rapidly by advertising associated attributes. However, the illustrations listed below are not meant to be restrictive or exhaustive.

- Advertising Service Functions and its attributes
  Service Function nodes implementing various service functions within the network need to advertise each service function they are offering so that a control and/or management entity can decide which instance to invoke for the delivery of an added-value service or to react to particular events (such as failure of a service function instance). Each service can be identified by a dedicated sub-TLV type while the associated attributes/identifiers of the service are indicated by the value part of the corresponding sub-TLV. These identifiers MAY not be globally unique and MAY not be exposed outside of a given administrative domain. The Self-defined sub-TLV Container TLV could appear multiple times within a given Router Information (RI) Opaque LSA, when more than one service function instances needs to be advertised by a given node based on a local policy.

  Advertising service functions and its attributes also allow the controller to adjust its policies and react dynamically. Typical actions would be, to withdraw a service instance from being invoked in the context of a service delivery, update load balancing polices, dynamically activate a backup instance, etc.

  The mechanisms, on how service information and attributes are used by an external controller (for example to steer the traffic) is beyond the scope of this document.

- Dissemination of dynamic information
  It’s possible for operators to disseminate the node local information like energy efficiency, congestion information, certain critical node statistics periodically to an external controller managing the network. How a Controller uses this information is beyond the scope of this document.
3. Terminology

This memo makes use of the terms defined in [RFC4970].

4. Self-defined Sub-TLV Container TLV

A new TLV within the body of the OSPFv2 and OSPFv3 RI Opaque LSA, called Self-defined Sub-TLV Container TLV is defined to carry one or more self-defined sub-TLVs.

The format of the Self-defined Sub-TLV Container TLV is shown in Figure 1.

```
+---------------------------------+---------------------------------+---------------------------------+
| Type                          | Length                          | First Self-defined Sub-TLV       |
|                               |                                 |                                 |
+---------------------------------+---------------------------------+---------------------------------+
// ...                            //                                 // ...
+---------------------------------+---------------------------------+---------------------------------+
| Last Self-defined Sub-TLV       |                                 |                                 |
|                               |                                 |                                 |
+---------------------------------+---------------------------------+---------------------------------+
```

**Figure 1: Self-defined Sub-TLV Container TLV**

Type: TBD Section 7

Length: A 16-bit field that indicates the length of the value portion in octets. It MUST be multiple of 4 octets dependent on the number of Self-defined Sub-TLVs advertised.

Value: Contains one or more nested TLV triplets of self-defined sub-TLVs as defined in Section 5.

There can be more than one TLV of these possible and the flooding scope of this TLV depends on the application. Being part of the RI Opaque LSA, the Self-defined sub-TLV Container TLV inherits applicability as well as restrictions as specified in Section 3 of [RFC4970].
5. Self-defined Sub-TLV

The self-defined sub-TLV has the following structure and can be part of the Container TLV as defined in Section 4 within the body of the OSPF RI LSA.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|        Attribute Length       |  Attribute Value (variable) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|        Attribute Length       |  Attribute Value (variable) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: Self-defined Sub-TLV

Type: Per Operator/Local Policy.

Length: A 16-bit field that indicates the length of the value portion in octets and will be padded/formatted as described in Section 2.1 of [RFC4970].

Value: Represents the associated attribute of the service or Type defined locally (i.e., within a single administrative domain). The Value field contains one or more (Attribute-Len, Attribute-value) tuple. Attribute Length is of 2 bytes, for fixed formatting and Attribute value as represented by attribute length.

The meaning of the self-defined sub-TLV is totally opaque to OSPF.

Routers advertising the self-defined sub-TLV are configured to do so without knowing (or even explicitly supporting) functionality implied by the sub-TLV.

The meaning of a self-defined sub-TLV is defined by the network local policy and is controlled via configuration.

How a receiving node communicates the self-defined sub-TLVs with the policy manager is outside the scope of this document.

There is no need for any specification to define any self-defined sub-TLV. Furthermore, the semantics of the self-defined sub-TLV order has no meaning. That is, there is no implied meaning to the ordering of the self-defined sub-TLV that indicates a certain operation or set of operations that need to be performed based on the
ordering. The ordering of self-defined sub-TLVs and the interpretation of the self-defined sub-TLV is deployment-specific. Routers can be configured with local policies if the order of sub-TLV must be preserved. How a router is configured with additional instructions (such as order preservation) is implementation-specific.

6. Acknowledgements

Authors would like to thank Acee Lindem for reviewing and providing suggestions on the initial version of the document. Also thankful to Anton Smirnov, Peter Psenak and Les Ginsberg for their review and comments.

7. IANA Considerations

This document includes a request to IANA to allocate a TLV type code for the new RI LSA TLV proposed in Section 4 of this document from OSPF Router Information (RI) TLVs Registry defined by [RFC4970].

8. Security Considerations

This document does not introduce any new security risk other than what is specified by [RFC4970].

9. References

9.1. Normative References


9.2. Informative References

Authors’ Addresses

Uma Chunduri
Ericsson Inc.
300 Holger Way,
San Jose, California 95134
USA
Phone: 408 750-5678
Email: uma.chunduri@ericsson.com

Xiaohu Xu
Huawei
Email: xuxiaohu@huawei.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain
Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://people.tid.es/LuisM.Contreras/

Mohamed Boucadair
France Telecom
Rennes 35000
France
Email: mohamed.boucadair@orange.com
Many OSPFv2 or OSPFv3 deployments run on overlay networks provisioned by means of pseudo-wires or L2-circuits. When the devices in the underlying network go for maintenance, it is useful to divert the traffic away from the node before the maintenance is actually scheduled. Since the nodes in the underlying network are not visible to OSPF, existing stub router mechanism described in [RFC3137] cannot be used. It is useful for routers in OSPFv2 or OSPFv3 routing domain to be able to advertise a link being in overload state to indicate impending maintenance activity in the underlying network devices.

This document describes the protocol extensions to disseminate link overload information in OSPFv2 and OSPFv3 protocol.

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
1.  Introduction

   It is useful for routers in OSPFv2 or OSPFv3 routing domain to be able to advertise a link being in overload state to indicate impending maintenance activity on the link. This document provides mechanisms to advertise link overload state in the flexible encodings.
provided by OSPFv2 Prefix/Link Attribute Advertisement ([I-D.ietf-ospf-prefix-link-attr]) and OSPFv3 Extended LSA ([I-D.ietf-ospf-ospfv3-lsa-extend]). Throughout this document, OSPF is used when the text applies to both OSPFv2 and OSPFv3. OSPFv2 or OSPFv3 is used when the text is specific to one version of the OSPF protocol.

2. Link overload sub TLV

2.1. OSPFv2 Link overload sub TLV

Link overload sub TLV is carried as part of the Extended link TLV as defined in [I-D.ietf-ospf-prefix-link-attr] for OSPFv2.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Remote IP address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 1: Link overload sub TLV for OSPFv2
```

Type : TBA
Length: 4
Value: Remote IPv4 address. The remote IP4 address is used to identify the particular link that is in overload state when there are multiple parallel links between two nodes.

2.2. OSPFv3 Link overload sub TLV

Link overload sub TLV is carried in the Router-link TLV as defined in the [I-D.ietf-ospf-ospfv3-lsa-extend] for OSPFv3. The Router-Link TLV contains the neighbor interface-id and can uniquely identify the link on the remote node.
Figure 2: Link overload sub TLV for OSPFv3

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA</td>
<td>0</td>
</tr>
</tbody>
</table>

3. Elements of procedure

The Link Overload sub TLV indicates that the link which carries the sub TLV is overloaded. The node that has the link going for maintenance, sets metric of the link to MAX-METRIC and re-originates the router LSA. The metric in the reverse direction also need to change to divert the traffic from reverse direction. The node SHOULD originate Link overload sub TLV and include it in Extended link TLV and originate the Extended Link Opaque LSA as defined in [I-D.ietf-ospf-prefix-link-attr] for OSPFv2 and E-Router-LSA as defined in [I-D.ietf-ospf-ospfv3-lsa-extend] for OSPFv3 and flood in the OSPF area.

when the originator of the Link Overload sub TLV, purges the extended link opaque LSA or re-originates without the Link Overload sub TLV, the metric on the remote node SHOULD be changed back to the original value.

Based on the link type of the overloaded link below actions MAY be taken by the receiver.

3.1. Point-to-point links

When a link overload TLV is received for a point-to-point link the receiver SHOULD identify the local link which corresponds to the overloaded link and set the metric to MAX-METRIC (0xffff). Receiver node MUST re-originate the router-LSA with the changed metric and flood into the OSPF area.

3.2. Broadcast/NBMA links

Broadcast or NBMA 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 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. For the Broadcast links, the MAX-METRIC on the outgoing link cannot be changed since all the neighbors are on same link. Setting the link cost to MAX-METRIC would impact paths going via all neighbors.

When a link-overload TLV is received by the remote end for a broadcast/NBMA link

- If it’s DROther or BDR for that link, SHOULD not take any action.
- If receiving node is DR for the link, it MUST remove the originator of the link overload TLV from the list of connected neighbors and MUST re-originate the network LSA and flood into the OSPF area.

3.3. Point-to-multipoint links

Operation for the point-to-multipoint links is similar to the point-to-point links. When a link overload TLV is received for a point-to-multipoint link the receiver SHOULD identify the neighbor which corresponds to the overloaded link and set the metric to MAX-METRIC (0xffffffff). Receiver node MUST re-originate the router-LSA with the changed metric and flood into the OSPF area.

4. Backward compatibility

The mechanism described in the document is fully backward compatible. It is required that the originator and receiver of link-overload sub TLV understand the extensions defined in this document and in case of broadcast links the originator and the DR need to understand the extensions. Other nodes in the network compute based on increased metric and hence the feature is backward compatible.

5. Applications

5.1. Pseudowire Services

```
-------------PE3--------------PE4-------------
       |                              |
       |                              |
CE1----------PE1----------------PE2--------CE2
                        -----------------------
                         Private VLAN
```

Figure 3: Pseudowire Services
Many service providers offer pseudo-wire services to customers using L2 circuits. The IGP protocol that runs in the customer network would also run over the pseudo-wire to get seamless private network for the customer. Service providers want to offer overload kind of functionality when the PE device is taken-out for maintenance. The provider should guarantee that the PE is taken out for maintenance only after the service is successfully diverted on the alternate path. Link overload feature provides facilities to achieve this service by increasing the metric on the link but still allowing the traffic to use the link when there is no alternate path available.

5.2. Controller based Traffic Engineering Deployments

![Diagram of Controller based Traffic Engineering](image)

Figure 4: Controller based Traffic Engineering

Controller based deployments where the controller participates in the IGP protocol gets the link-overload information when the link maintenance is impending. Using this information controller finds an alternate path. If there are no alternate paths satisfying the traffic engineering constraints, controller might temporarily relax the constraints and put the service on different path. In the above example when P1->P2 link goes for maintenance, controller gets the link-overload information and sets up an alternate path via P1->P3->P2. Once the traffic is diverted, P1->P2 link can be taken out for maintenance/upgrade.

6. Security Considerations

This document does not introduce any further security issues other than those discussed in [RFC2328] and [RFC5340].
7. IANA Considerations

This specification updates one OSPF registry:

OSPF Extended Link TLVs Registry

i) TBD - Link Overload TLV
OSPFV3 Router Link TLV Registry

i) TBD - Link Overload TLV

8. Acknowledgements

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses
OSPFv2 Extensions for BIER
draft-ietf-bier-ospf-bier-extensions-18.txt

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain multicast related per-flow state. Neither does BIER require an explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. Such header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by the according set of bits set in BIER packet header.

This document describes the OSPF [RFC2328] protocol extension required for BIER with MPLS encapsulation [RFC8296]. Support for other encapsulation types is outside the scope of this document. The use of multiple encapsulation types is outside the scope of this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.
1. Introduction

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. Neither does BIER explicitly require a tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast...
packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

BIER architecture requires routers participating in BIER to exchange BIER related information within a given domain. BIER architecture permits link-state routing protocols to perform distribution of such information. This document describes extensions to OSPF necessary to advertise BIER specific information in the case where BIER uses MPLS encapsulation as described in [RFC8296].

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. Flooding of the BIER Information in OSPF

All BIER specific information that a Bit-Forwarding Router (BFR) needs to advertise to other BFRs is associated with a BFR-Prefix. A BFR prefix is a unique (within a given BIER domain) routable IP address that is assigned to each BFR as described in more detail in section 2 of [RFC8279].

Given that BIER information must be associated with a BFR prefix, the OSPF Extended Prefix Opaque LSA [RFC7684] has been chosen for advertisement.

2.1. BIER Sub-TLV

A Sub-TLV of the Extended Prefix TLV (defined in [RFC7684]) is defined for distributing BIER information. The Sub-TLV is called the BIER Sub-TLV. Multiple BIER Sub-TLVs may be included in the Extended Prefix TLV.

The BIER 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
+-+-+-+-+-+-+-+---------------+---------------+
|              Type             |             Length            |
+-+-+-+-+-+-+-+---------------+---------------+
| Sub-domain-ID |     MT-ID     |              BFR-id           |
+-+-+-+-+-+-+-+---------------+---------------+
|    BAR        |    IPA        |            Reserved           |
+-+-+-+-+-+-+-+---------------+---------------+
|                      Sub-TLVs (variable) |
+-+-+-+-+-+-+-+---------------+---------------+
```

Psenak, et al. Expires December 3, 2018
Type: 9

Length: Variable, dependent on sub-TLVs.

Sub-domain-ID: Unique value identifying the BIER sub-domain within the BIER domain, as described in section 1 of [RFC8279].

MT-ID: Multi-Topology ID (as defined in [RFC4915]) that identifies the topology that is associated with the BIER sub-domain.

BFR-id: A 2 octet field encoding the BFR-id, as documented in section 2 of [RFC8279]. If the BFR is not locally configured with a valid BFR-id, the value of this field is set to 0, which is defined as illegal in [RFC8279].

BAR: Single octet BIER specific algorithm used to calculate underlay paths to reach other BFRs. Values are allocated from the "BIER Algorithm Registry" which is defined in [I-D.ietf-bier-isis-extensions].

IPA: Single octet IGP algorithm to either modify, enhance or replace the calculation of underlay paths to reach other BFRs as defined by the BAR value. Values are defined in the "IGP Algorithm Types" registry.

Each BFR sub-domain MUST be associated with one and only one OSPF topology that is identified by the MT-ID. If the association between BIER sub-domain and OSPF topology advertised in the BIER sub-TLV by other BFRs is in conflict with the association locally configured on the receiving router, the BIER Sub-TLV MUST be ignored.

If the MT-ID value is outside of the values specified in [RFC4915], the BIER Sub-TLV MUST be ignored.

If a BFR advertises the same Sub-domain-ID in multiple BIER sub-TLVs, the BFR MUST be treated as if it did not advertise a BIER sub-TLV for such sub-domain.

All BFRs MUST detect advertisement of duplicate valid BFR-IDs for a given MT-ID and Sub-domain-ID. When such duplication is detected by the BFR, it MUST behave as described in section 5 of [RFC8279].

The supported BAR and IPA algorithms MUST be consistent for all routers supporting a given BFR sub-domain. A router receiving BIER Sub-TLV advertisement with a value in BAR or IPA fields which does not match the locally configured value for a given BFR sub-domain, MUST report a misconfiguration for such BIER sub-domain and MUST ignore such BIER sub-TLV.
The use of non-zero values in either the BAR field or the IPA field is outside the scope of this document.

2.2.  BIER MPLS Encapsulation Sub-TLV

The BIER MPLS Encapsulation Sub-TLV is a Sub-TLV of the BIER Sub-TLV. The BIER MPLS Encapsulation Sub-TLV is used in order to advertise MPLS specific information used for BIER. It MAY appear multiple times in the BIER Sub-TLV.

The BIER MPLS Encapsulation Sub-TLV has the following format:

```
+--------------------------------+------------------+
|              Type              |     Length      |
+--------------------------------+------------------+
|     Max SI     |                     Label                     |
+--------------------------------+------------------+
|      BS Len    |     Reserved          |
+--------------------------------+------------------+
```

Type: 10
Length: 8 octets
Max SI : A 1 octet field encoding the maximum Set Identifier (section 1 of [RFC8279]), used in the encapsulation for this BIER sub-domain for this bitstring length.
Label: A 3 octet field, where the 20 rightmost bits represent the first label in the label range. The 4 leftmost bits MUST be ignored.
Bit String Length: A 4 bits field encoding the supported BitString length associated with this BFR-prefix. The values allowed in this field are specified in section 2 of [RFC8296].
Reserved: SHOULD be set to 0 on transmission and MUST be ignored on reception.

The "label range" is the set of labels beginning with the Label and ending with (Label + (Max SI)). A unique label range is allocated for each BitString length and Sub-domain-ID. These labels are used for BIER forwarding as described in [RFC8279] and [RFC8296].
The size of the label range is determined by the number of Set Identifiers (SI) (section 1 of [RFC8279]) that are used in the network. Each SI maps to a single label in the label range. The first label is for SI=0, the second label is for SI=1, etc.

If the label associated with the Maximum Set Identifier exceeds the 20 bit range, the BIER MPLS Encapsulation Sub-TLV MUST be ignored.

If the BS length is set to a value that does not match any of the allowed values specified in [RFC8296], the BIER MPLS Encapsulation Sub-TLV MUST be ignored.

If same BS length is repeated in multiple BIER MPLS Encapsulation Sub-TLV inside the same BIER Sub-TLV, the BIER sub-TLV MUST be ignored.

Label ranges within all BIER MPLS Encapsulation Sub-TLVs advertised by the same BFR MUST NOT overlap. If the overlap is detected, the advertising router MUST be treated as if it did not advertise any BIER sub-TLVs.

2.3. Flooding scope of BIER Information

The flooding scope of the OSPF Extended Prefix Opaque LSA [RFC7684] that is used for advertising the BIER Sub-TLV is set to area-local. To allow BIER deployment in a multi-area environment, OSPF must propagate BIER information between areas.

```
(   ) (   ) (   ) (   )
   (   ) (   ) (   ) (   )
R1  Area 1  R2  Area 0  R3  Area 2  R4
   (   ) (   ) (   ) (   )
   (   ) (   ) (   ) (   )
```

Figure 1: BIER propagation between areas

The following procedure is used in order to propagate BIER related information between areas:

When an OSPF Area Border Router (ABR) advertises a Type-3 Summary LSA from an intra-area or inter-area prefix to all its attached 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. The route-type in the OSPF Extended Prefix TLV is set to inter-area. When determining
whether a BIER Sub-TLV should be included in this LSA, an OSPF ABR will:

- Examine its best path to the prefix in the source area and find the advertising router associated with the best path to that prefix.

- Determine if such advertising router advertised a BIER Sub-TLV for the prefix. If yes, the ABR will copy the information from such BIER Sub-TLV when advertising BIER Sub-TLV to each attached area.

In the Figure 1, R1 advertises a prefix 192.0.2.1/32 in Area 1. It also advertises Extended Prefix Opaque LSA for prefix 192.0.2.1/32 and includes BIER Sub-TLV in it. Area Border Router (ABR) R2 calculates the reachability for prefix 192.0.2.1/32 inside Area 1 and propagates it to Area 0. When doing so, it copies the entire BIER Sub-TLV (including all its Sub-TLVs) it received from R1 in Area 1 and includes it in the Extended Prefix Opaque LSA it generates for 192.0.2.1/32 in Area 0. ABR R3 calculates the reachability for prefix 192.0.2.1/32 inside Area 0 and propagates it to Area 2. When doing so, it copies the entire BIER Sub-TLV (including all its Sub-TLVs) it received from R2 in Area 0 and includes it in the Extended Prefix Opaque LSA it generates for 192.0.2.1/32 in Area 2.

3. Security Considerations

This document introduces new sub-TLVs for existing OSPF Extended Prefix TLV. It does not introduce any new security risks to OSPF. Existing security extensions as described in [RFC2328] and [RFC7684] apply.

It is assumed that both BIER and OSPF layer is under a single administrative domain. There can 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.

The Security Considerations section of [RFC8279] discusses the possibility of performing a Denial of Service (DoS) attack by setting too many bits in the BitString of a BIER-encapsulated packet. However, this sort of DoS attack cannot be initiated by modifying the OSPF BIER advertisements specified in this document. A BFIR decides which systems are to receive a BIER-encapsulated packet. In making this decision, it is not influenced by the OSPF control messages. When creating the encapsulation, the BFIR sets one bit in the encapsulation for each destination system. The information in the
OSPF BIER advertisements is used to construct the forwarding tables that map each bit in the encapsulation into a set of next hops for the host that is identified by that bit, but is not used by the BFIR to decide which bits to set. Hence an attack on the OSPF control plane cannot be used to cause this sort of DoS attack.

While a BIER-encapsulated packet is traversing the network, a BFR that receives a BIER-encapsulated packet with n bits set in its BitString may have to replicate the packet and forward multiple copies. However, a given bit will only be set in one copy of the packet. That means that each transmitted replica of a received packet has fewer bits set (i.e., is targeted to fewer destinations) than the received packet. This is an essential property of the BIER forwarding process as defined in [RFC8279]. While a failure of this process might cause a DoS attack (as discussed in the Security Considerations of [RFC8279]), such a failure cannot be caused by an attack on the OSPF control plane.

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 OSPF 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.

4. IANA Considerations

The document requests two new allocations from the OSPF Extended Prefix sub-TLV registry as defined in [RFC7684].

- BIER Sub-TLV: 9
- BIER MPLS Encapsulation Sub-TLV: 10

5. Acknowledgments

The authors would like to thank Rajiv Asati, Christian Martin, Greg Shepherd and Eric Rosen for their contribution.

6. Normative References

[I-D.ietf-bier-isis-extensions]
Internet-Draft OSPFv2 Extensions for BIER June 2018


Authors' Addresses

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

Email: ppsenak@cisco.com
Sam Aldrin
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
USA

Email: aldrin.ietf@gmail.com
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 OSPFv3 extensions required for Segment Routing with MPLS data plane.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 13, 2019.
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 - OSPFv3 extension for SR with IPv6 data plane will be specified in a separate document. 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.

This draft describes the OSPFv3 extensions required for Segment Routing with MPLS data plane.

Segment Routing architecture is described in [RFC8402].

Segment Routing use cases are described in [RFC7855].

2. Terminology

This section lists some of the terminology used in this document:

- ABR - Area Border Router
- Adj-SID - Adjacency Segment Identifier
- AS - Autonomous System
- ASBR - Autonomous System Boundary Router
- DR - Designated Router
- IS-IS - Intermediate System to Intermediate System
- LDP - Label Distribution Protocol
- LSP - Label Switched Path
- MPLS - Multi Protocol Label Switching
3. Segment Routing Identifiers

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

3.1. 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:

```
+---------------+---------------+---------------+---------------+
|                |                |                |                |
|                |                |                |                |
|                |  Type         |  Length       |                |
|                |                |                |                |
|                |  SID/Label    |                |                |
|                |                |                |                |
```

where:

Type: 7

Length: Either 3 or 4 octets

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 than 3 or 4.

4. 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 OSPFv3 Router Information Opaque LSA (defined in [RFC7770]) and specified in [I-D.ietf-ospf-segment-routing-extensions].

5. OSPFv3 Extended Prefix Range TLV

In some cases it is useful to advertise attributes for a range of prefixes in a single advertisement. The Segment Routing Mapping Server, which is described in [I-D.ietf-spring-segment-routing-ldp-interop], is an example of where SIDs for multiple prefixes can be advertised. To optimize such advertisement in case of multiple prefixes from a contiguous address range, OSPFv3 Extended Prefix Range TLV is defined.

The OSPFv3 Extended Prefix Range TLV is a top-level TLV of the following LSAs defined in [RFC8362]:

   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 each LSA mentioned above. The OSPFv3 Extended Prefix Range TLV has the following format:
where:

Type: 9

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

Prefix length: Length of prefix in bits.

AF: Address family for the prefix.

AF: 0 - IPv4 unicast
AF: 1 - 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 the IPv4 multicast address range (224.0.0.0/3), if the AF is IPv4 unicast

Addresses other than the IPv6 unicast addresses, if the AF is IPv6 unicast

Flags: Reserved. MUST be zero when sent and are ignored when received.

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

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.

For the address family IPv6 unicast, the prefix, encoded as an even multiple of 32-bit words, padded with zeroed bits as necessary. This encoding consumes \(((\text{PrefixLength} + 31) / 32)\) 32-bit words.

Prefix encoding for other address families is beyond the scope of this specification. Prefix encoding for other address families can be defined in the future standard-track IETF specifications.

The range represents the contiguous set of prefixes with the same prefix length as specified by the Prefix Length field. The set starts with the prefix that is specified by the Address Prefix field. The number of prefixes in the range is equal to the Range size.

If the OSPFv3 Extended Prefix Range TLVs advertising the exact same range appears in multiple LSAs of the same type, originated by the same OSPFv3 router, the LSA with the numerically smallest Instance ID MUST be used and subsequent instances of the OSPFv3 Extended Prefix Range TLVs MUST be ignored.

6. Prefix SID Sub-TLV

The Prefix SID Sub-TLV is a Sub-TLV of the following OSPFv3 TLVs as defined in [RFC8362] and in Section 5:

- 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:
where:

Type: 4

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, 2 for IPv6) 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.
Reserved: SHOULD be set to 0 on transmission and MUST be ignored on reception.

Algorithm: Single octet identifying the algorithm the Prefix-SID is associated with as defined in [I-D.ietf-ospf-segment-routing-extensions].

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 [I-D.ietf-ospf-segment-routing-extensions] MUST ignore the Prefix-SID Sub-TLV.

SID/Index/Label: According to the V-Flag and L-Flag, it contains:

- V-flag is set to 0 and L-flag is set to 0: The SID/Index/Label field is a 4 octet index defining the offset in the SID/Label space advertised by this router.
- V-flag is set to 1 and L-flag is set to 1: The SID/Index/Label field is a 3 octet local label where the 20 rightmost bits are used for encoding the label value.
- All other combinations of V-flag and L-flag are invalid and any SID advertisement received with an invalid setting for V and L flags MUST be ignored.

If an OSPFv3 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 prefixes that are propagated between areas by an ABR based on intra-area or inter-area reachability, unless the advertised prefix is directly attached to such 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 advertising 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 needs to stitch the incoming packet into a continuing MPLS LSP to the final destination. This could occur at an ABR (prefix propagation from one area to another) or at an ASBR (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 Traffic Class field [RFC5462].

When the M-Flag is set, the NP-flag and the E-flag MUST be ignored on 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 the downstream neighbor is an ABR, which is advertising prefix reachability and is setting the LA-bit in the Prefix Options as described in [RFC8362].

The Prefix is external type and the downstream neighbor is an ASBR, which is advertising prefix reachability and is setting the LA-bit in the Prefix Options as described in [RFC8362].

When a Prefix-SID is advertised in the OSPFv3 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: 2001:DB8::1/128, Prefix-SID: Index 1
Router-B: 2001:DB8::2/128, Prefix-SID: Index 2
Router-C: 2001:DB8::3/128, Prefix-SID: Index 3
Router-D: 2001:DB8::4/128, Prefix-SID: Index 4
then the Address Prefix field in the OSPFv3 Extended Prefix Range TLV would be set to 2001:DB8::1, the Prefix Length would be set to 128, the 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:

- 2001:DB8:1::0/120, Prefix-SID: Index 51
- 2001:DB8:1::100/120, Prefix-SID: Index 52
- 2001:DB8:1::200/120, Prefix-SID: Index 53
- 2001:DB8:1::300/120, Prefix-SID: Index 54
- 2001:DB8:1::400/120, Prefix-SID: Index 55
- 2001:DB8:1::500/120, Prefix-SID: Index 56
- 2001:DB8:1::600/120, Prefix-SID: Index 57

then the Prefix field in the OSPFv3 Extended Prefix Range TLV would be set to 2001:DB8:1::0, the Prefix Length would be set to 120, the Range Size would be set to 7, and the Index value in the Prefix-SID Sub-TLV would be set to 51.

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 Adj-SID Sub-TLV is an optional Sub-TLV of the Router-Link TLV as defined in [RFC8362]. It MAY appear multiple times in the Router-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         |     Weight    |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

where:

- **Type**: 5
- **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 [RFC8402].

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 the same across router restart and/or interface flap.

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

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

Weight: Weight used for load-balancing purposes. The use of the weight is defined in [RFC8402].

SID/Index/Label: as described in Section 6.

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 [RFC8402].

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 Sub-TLV 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 router on a broadcast, NBMA, or hybrid [RFC6845] network.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flags</td>
<td>Weight</td>
</tr>
<tr>
<td>Neighbor ID</td>
<td></td>
</tr>
<tr>
<td>SID/Index/Label</td>
<td>(variable)</td>
</tr>
</tbody>
</table>

where:

Type: 6

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

Flags: same as in Section 7.1

Weight: Weight used for load-balancing purposes. The use of the weight is defined in [RFC8402].

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

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

SID/Index/Label: as described in Section 6.

When the P-flag is not set, the LAN Adj-SID MAY be persistent. When the P-flag is set, the LAN 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 to which it is advertising reachability (e.g., a loopback IP address as described in Section 6).

A Prefix-SID can also be advertised by 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 OSPFv3 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 SR Mapping Server could use either area flooding scope or autonomous system flooding scope when advertising Prefix SIDs for prefixes, based on the configuration of the SR Mapping Server. Depending on the flooding scope used, the SR Mapping Server chooses the OSPFv3 LSA type that will be used. If the area flooding scope is needed, an E-Intra-Area-Prefix-LSA [RFC8362] is used. If autonomous system flooding scope is needed, an E-AS-External-LSA [RFC8362] 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 6), 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 an Inter-Area Prefix TLV, External-Prefix TLV, or Intra-Area-Prefix TLV [RFC8362] does not itself contribute to the prefix reachability. The NU-bit [RFC5340] 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 from contributing to prefix reachability.

An SR Mapping Server MUST use the OSPFv3 Extended Prefix Range TLVs when advertising SIDs for prefixes. Prefixes of different route-types can be combined in a single OSPFv3 Extended Prefix Range TLV advertised by an SR Mapping Server.

Area-scoped OSPFv3 Extended Prefix Range TLVs are propagated between areas, similar to propagation of prefixes between areas. Same rules that are used for propagating prefixes between areas [RFC5340] are used for the propagation of the prefix ranges.
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 to propagate Prefix SIDs between areas.

When an OSPFv3 ABR advertises an Inter-Area-Prefix-LSA from an intra-area prefix to all its connected areas, it will also include the Prefix-SID Sub-TLV, as described in Section 6. 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 OSPFv3 ABR advertises Inter-Area-Prefix-LSA LSAs from an inter-area route to all its connected areas, it will also include the Prefix-SID Sub-TLV, as described in Section 6. The Prefix-SID value 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.
8.3. Segment Routing for External Prefixes

AS-External-LSAs are flooded domain wide. When an ASBR, which supports SR, originates an E-AS-External-LSA, it SHOULD also include a Prefix-SID Sub-TLV, as described in Section 6. The Prefix-SID value will be set to the SID that has been reserved for that prefix.

When an NSSA [RFC3101] ABR 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 for 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 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 than 2-Way, then the Adj-SID advertisement MUST be withdrawn from the area.

8.4.2. Adjacency SID on Broadcast or NBMA Interfaces

Broadcast, NBMA, or hybrid [RFC6845] networks in OSPFv3 are represented by a star topology where the 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 7.1.

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 7.2.
9. IANA Considerations

This specification updates several existing OSPFv3 registries.

9.1. OSPFv3 Extended-LSA TLV Registry

Following values are allocated:

- 9 - OSPFv3 Extended Prefix Range TLV

9.2. OSPFv3 Extended-LSA Sub-TLV registry

- 4 - Prefix SID Sub-TLV
- 5 - Adj-SID Sub-TLV
- 6 - LAN Adj-SID Sub-TLV
- 7 - SID/Label Sub-TLV

10. Security Considerations

With the OSPFv3 segment routing extensions defined herein, OSPFv3 will now program the MPLS data plane [RFC3031]. 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 can be more difficult to detect and isolate.

Existing security extensions as described in [RFC5340] and [RFC8362] apply to these segment routing extensions. While OSPFv3 is under a single administrative domain, there can be deployments where potential attackers have access to one or more networks in the OSPFv3 routing domain. In these deployments, stronger authentication mechanisms such as those specified in [RFC4552] or [RFC7166] 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 OSPFv3 router or routing process. Reception of a 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 OSPFv3 control plane.
11. Contributors

The following people gave a substantial contribution to the content of this document and should be considered as co-authors:

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  
Nuage Networks  
US  

Email: jefftant.ietf@gmail.com

Thanks to Acee Lindem for his substantial contribution to the content of this document.

We would like to thank Anton Smirnov for his contribution as well.
12. References

12.1. Normative References


Internet-Draft OSPFv3 Extensions for Segment Routing January 2019


12.2. Informative References


Authors’ Addresses

Peter Psenak (editor)
Cisco Systems, Inc.
Eurovea Centre, Central 3
Pribinova Street 10
Bratislava 81109
Slovakia

Email: ppsenak@cisco.com

Stefano Previdi (editor)
Individual

Email: stefano.previdi@net
OSPF Extensions for Segment Routing
draft-ietf-ospf-segment-routing-extensions-27

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 OSPFv2 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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on June 6, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the
document authors. All rights reserved.

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

Table of Contents

1.  Introduction ................................................... 3
2.  Segment Routing Identifiers .................................... 3
  2.1.  SID/Label Sub-TLV ........................................... 4
3.  Segment Routing Capabilities ................................... 4
  3.1.  SR-Algorithm TLV ........................................... 4
  3.2.  SID/Label Range TLV ......................................... 6
  3.3.  SR Local Block TLV .......................................... 8
  3.4.  SRMS Preference TLV ......................................... 10
4.  OSPF Extended Prefix Range TLV ................................ 11
5.  Prefix SID Sub-TLV .............................................. 13
6.  Adjacency Segment Identifier (Adj-SID) ........................ 16
  6.1.  Adj-SID Sub-TLV ............................................. 17
  6.2.  LAN Adj-SID Sub-TLV ......................................... 18
7.  Elements of Procedure .......................................... 19
  7.1.  Intra-area Segment routing in OSPFv2 ....................... 19
  7.2.  Inter-area Segment routing in OSPFv2 ....................... 20
  7.3.  Segment Routing for External Prefixes ..................... 21
  7.4.  Advertisement of Adj-SID ................................... 22
    7.4.1.  Advertisement of Adj-SID on Point-to-Point Links .... 22
    7.4.2.  Adjacency SID on Broadcast or NBMA Interfaces ...... 22
8.  IANA Considerations ............................................ 22
  8.1.  OSPF Router Information (RI) TLVs Registry ............... 22
  8.2.  OSPFv2 Extended Prefix Opaque LSA TLVs Registry .......... 23
  8.3.  OSPFv2 Extended Prefix Sub-TLVs Registry ................ 23
  8.4.  OSPFv2 Extended Link TLV Sub-TLVs Registry ................ 23
  8.5.  IGP Algorithm Type Registry ................................ 23
9.  Implementation Status .......................................... 24
10. Security Considerations ........................................ 25
11. Contributors .................................................. 26
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. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              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]). The TLVs defined below are applicable to both OSPFv2 and OSPFv3; see also [I-D.ietf-ospf-ospfv3-segment-routing-extensions]

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 can 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
<table>
<thead>
<tr>
<th>Algorithm 1</th>
<th>Algorithm...</th>
<th>Algorithm n</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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 MUST 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 MUST 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 MUST be used and subsequent instances of the SR-Algorithm TLV MUST 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              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.

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

- The originating router MUST encode each range into a different SID/Label Range TLV.
- 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.
- The receiving router MUST adhere to the order in which the ranges are advertised when calculating a SID/label from a SID index.
- The originating router MUST NOT advertise overlapping ranges.
- When a router receives multiple overlapping ranges, it MUST conform to the procedures defined in [I-D.ietf-spring-segment-routing-mpls].

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:

\[
\text{SRGB} = [100, 199] \\
[1000, 1099] \\
[500, 599]
\]

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 can instruct the router to allocate a specific local SID. Some controllers or applications can use the control plane to discover the available set of local SIDs on a particular router. In such cases, the SRLB 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:
<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Variable, in octets, dependent on Sub-TLVs.</td>
</tr>
</tbody>
</table>

**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.

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

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 than 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 can 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-segment-routing-ldp-interop].

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Preference    |                 Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: 15
- **Length**: 4 octets
- **Preference**: 1 octet. SRMS preference value from 0 to 255.
- **Reserved**: SHOULD be set to 0 on transmission and MUST be ignored on reception.

When multiple SRMS Preference TLVs are received from a given router, the receiver MUST 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 MUST 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 MUST be used and subsequent instances of the SRMS Preference TLV MUST 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 than 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              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.

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

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

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:

- V-flag is set to 0 and L-flag is set to 0: The SID/Index/Label field is a 4 octet index defining the offset in the SID/Label space advertised by this router.

- V-flag is set to 1 and L-flag is set to 1: The SID/Index/Label field is a 3 octet local label where the 20 rightmost bits are used for encoding the label value.

All other combinations of V-flag and L-flag are invalid and any SID advertisement received with an invalid setting for V and L flags MUST be ignored.

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 need to 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:

```
+---------------------------------------------------------------+
|                   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:

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

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.

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

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: as described in Section 5.

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.
where:

Type: 3

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

Flags: same as in Section 6.1

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

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: as described in Section 5.

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 PrefixOpaque LSA, as described in [RFC7684]. The flooding scope of the Extended PrefixOpaque 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 PrefixOpaque LSAs, as described in [RFC7684]. The flooding scope of the Extended PrefixOpaque 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 [RFC3101] 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 than 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 Router Information (RI) TLVs Registry

- 8 (IANA Preallocated) - SR-Algorithm TLV
- 9 (IANA Preallocated) - SID/Label Range TLV
- 14 - SR Local Block TLV
- 15 - SRMS Preference TLV
8.2. OSPFv2 Extended Prefix Opaque LSA TLVs Registry

Following values are allocated:

- 2 - OSPF Extended Prefix Range TLV

8.3. OSPFv2 Extended Prefix TLV Sub-TLVs Registry

Following values are allocated:

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

8.4. OSPFv2 Extended Link TLV Sub-TLVs Registry

Following initial values are allocated:

- 1 - SID/Label Sub-TLV
- 2 - Adj-SID Sub-TLV
- 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" under a new category of "Interior Gateway Protocol (IGP) Parameters" IANA registries. The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

Values in this registry 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:

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 interoperability 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 can 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 can 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]

[I-D.ietf-spring-segment-routing-ldp-interop]

[I-D.ietf-spring-segment-routing-mpls]


13.2. Informative References

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]

[RFC7474]

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
Apstra, Inc.
Email: jefftant.ietf@gmail.com
Abstract

This document defines a YANG data model that can be used to configure and manage OSPF. The model is based on YANG 1.1 as defined in RFC 7950 and conforms to the Network Management Datastore Architecture (NMDA) as described in RFC 8342.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 19, 2020.

Copyright Notice

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

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

Table of Contents

1. Overview ......................................................... 2
   1.1. Requirements Language ..................................... 3
   1.2. Tree Diagrams ............................................... 3
2. Design of Data Model ........................................... 3
   2.1. OSPF Operational State ..................................... 3
   2.2. Overview .................................................... 4
   2.3. OSPFv2 and OSPFv3 .......................................... 5
   2.4. Optional Features .......................................... 5
   2.5. OSPF Router Configuration/Operational State ............... 7
   2.6. OSPF Area Configuration/Operational State ................ 10
   2.7. OSPF Interface Configuration/Operational State .......... 16
   2.8. OSPF Notifications ......................................... 19
   2.9. OSPF RPC Operations ....................................... 23
3. OSPF YANG Module ................................................ 23
4. Security Considerations ........................................ 120
5. IANA Considerations ............................................. 123
6. Acknowledgements ................................................ 123
7. References ....................................................... 124
   7.1. Normative References ...................................... 124
   7.2. Informative References .................................... 129
App. A. Contributors’ Addresses ................................. 131
Authors’ Addresses ................................................ 131

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], RESTCONF [RFC8040], and other Network Management protocols. 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. It fully conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. A core routing data model is defined in [RFC8349], and it provides the basis for the development of data models for routing protocols. The interface data model is defined in [RFC8343] 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 OSPFv2 or OSPFv3 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Tree Diagrams

This document uses the graphical representation of data models defined in [RFC8340].

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 instance is tied to the routing domain and how multiple protocol instances 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 the Network Management Datastore Architecture [RFC8342]. Consequently, only the routing container in the ietf-routing model [RFC8349] 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. The ietf-ospf model defines a single instance of OSPF which may be instantiated as an OSPFv2 or OSPFv3 instance. Multiple instances are instantiated as multiple control-plane-protocols instances.

module: ietf-ospf
  augment /rt:routing/rt:control-plane-protocols/
  rt:control-plane-protocol:
    +++-rw ospf
      
      +++-rw af? identityref
        
        +++-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 container includes one OSPF protocol instance. The instance includes OSPF router level configuration and operational state. Each OSPF instance maps to a control-plane-protocol instance as defined in [RFC8349].
The area and area/interface containers define the OSPF configuration and operational state for OSPF areas and interfaces respectively.

The topologies 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 the following optional features:

1. multi-topology: Support Multi-Topology Routing (MTR) [RFC4915].
2. multi-area-adj: Support OSPF multi-area adjacency [RFC5185].
4. demand-circuit: Support OSPF demand circuits [RFC1793].
5. mtu-ignore: Support disabling OSPF Database Description packet MTU mismatch checking specified in section 10.6 of [RFC2328].
6. lls: Support OSPF link-local signaling (LLS) [RFC5613].
7. prefix-suppression: Support OSPF prefix advertisement suppression [RFC6860].
8. ttl-security: Support OSPF Time to Live (TTL) security check support [RFC5082].
9. nsr: Support OSPF Non-Stop Routing (NSR). The OSPF NSR feature allows a router with redundant control-plane capability (e.g., dual Route-Processor (RP) cards) to maintain its state and adjacencies during planned and unplanned control-plane processing restarts. It differs from graceful-restart or Non-
Stop Forwarding (NSF) in that no protocol signaling or assistance from adjacent OSPF neighbors is required to recover control-plane state.

10. graceful-restart: Support Graceful OSPF Restart [RFC3623], [RFC5187].

11. auto-cost: Support OSPF interface cost calculation according to reference bandwidth [RFC2328].

12. max-ecmp: Support configuration of the maximum number of Equal-Cost Multi-Path (ECMP) paths.

13. max-lsa: Support configuration of the maximum number of LSAs the OSPF instance will accept [RFC1765].

14. te-rid: Support configuration of the Traffic Engineering (TE) Router-ID, i.e., the Router Address described in Section 2.4.1 of [RFC3630] or the Router IPv6 Address TLV described in Section 3 of [RFC5329].

15. ldp-igp-sync: Support LDP IGP synchronization [RFC5443].

16. ospfv2-authentication-trailer: Support OSPFv2 Authentication trailer as specified in [RFC5709] or [RFC7474].

17. ospfv3-authentication-ipsec: Support IPsec for OSPFv3 authentication [RFC4552].

18. ospfv3-authentication-trailer: Support OSPFv3 Authentication trailer as specified in [RFC7166].

19. fast-reroute: Support IP Fast Reroute (IP-FRR) [RFC5714].

20. node-flag: Support node-flag for OSPF prefixes. [RFC7684].

21. node-tag: Support node admin tag for OSPF instances [RFC7777].

22. lfa: Support Loop-Free Alternates (LFAs) [RFC5286].

23. remote-lfa: Support Remote Loop-Free Alternates (R-LFA) [RFC7490].

24. stub-router: Support RFC 6987 OSPF Stub Router advertisement [RFC6987].

25. pe-ce-protocol: Support OSPF as a PE-CE protocol [RFC4577], [RFC6565].

27. bfd: Support BFD detection of OSPF neighbor reachability
    [RFC5880], [RFC5881], and [I-D.ietf-bfd-yang].

28. hybrid-interface: Support OSPF Hybrid Broadcast and Point-to-
    Point Interfaces [RFC6845].

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
represents an OSPF protocol instance and contains the router level
configuration and operational state. The operational state includes
the instance 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 af iana-rt-types:address-family
      +--rw enable? boolean
      +--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 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
    ++--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
++--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)?
++--ro protected-routes
  ++--ro af-stats* [af prefix alternate]
    ++--ro af       iana-rt-types:address-family
    ++--ro prefix   string
    ++--ro alternate string
    ++--ro alternate-type? enumeration
    ++--ro best?    boolean
    ++--ro non-best-reason? string
    ++--ro protection-available? bits
    ++--ro alternate-metric1?  uint32
    ++--ro alternate-metric2?  uint32
    ++--ro alternate-metric3?  uint32
++--ro unprotected-routes
  ++--ro af-stats* [af prefix]
    ++--ro af       iana-rt-types:address-family
    ++--ro prefix   string
++--ro protection-statistics* [frr-protection-method]
+--ro frr-protection-method string
+-ro af-stats* [af]
  +--ro af iana-rt-types:address-family
  +--ro total-routes? uint32
  +--ro unprotected-routes? uint32
  +--ro protected-routes? uint32
  +--ro linkprotected-routes? uint32
  +--ro nodeprotected-routes? uint32
+-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
    +--rw 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 discontinuity-time yang:date-and-time
      +--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-cksum-sum? uint32
    +--ro database
      +--ro as-scope-lsa-type* [lsa-type]
        +--ro lsa-type? uint16
        +--ro lsa-count? yang:gauge32
        +--ro lsa-cksum-sum? int32
    +--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
### 2.6. 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).

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

            ---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
++rw topologies {ospf:multi-topology}? 
  ++rw topology* [name]
    ++rw name -> ../../../../../../../../
    ++rw summary?        boolean
    ++rw default-cost?   ospf-metric
    ++rw ranges
      ++rw range* [prefix]
        ++rw prefix       inet:ip-prefix
        ++rw advertise?   boolean
        ++rw cost?        ospf-metric
++ro statistics
  ++ro discontinuity-time           yang:date-and-time
  ++ro spf-runs-count?              yang:counter32
  ++ro abr-count?                   yang:gauge32
  ++ro asbr-count?                   yang:gauge32
  ++ro ar-nssa-translator-event-count?
    ++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
  +++rw authentication
    +++rw (auth-type-selection)?
      +++:(ospfv2-auth)
        +++rw ospfv2-auth-trailer-rfc?
          ospfv2-auth-trailer-rfc-version
        +++rw (ospfv2-auth-specification)?
          +++:(auth-key-chain) (key-chain)?
            +++rw ospfv2-key-chain? key-chain:key-chain-ref
            +++:(auth-key-explicit)
            +++rw ospfv2-key-id?    uint32
            +++rw ospfv2-key?       string
            +++rw ospfv2-crypto-algorithm?
              identityref
        +++:(ospfv3-auth-ipsec)
          +++rw sa?               string
        +++:(ospfv3-auth-trailer)
          +++rw ospfv3-authentication-trailer??
        +++rw (ospfv3-auth-specification)?
          +++:(auth-key-chain) (key-chain)?
            +++rw ospfv3-key-chain? key-chain:key-chain-ref
            +++:(auth-key-explicit)
            +++rw ospfv3-sa-id?      uint16
            +++rw ospfv3-key?        string
            +++rw ospfv3-crypto-algorithm?
              identityref
  +++ro cost?              uint16
  +++ro state?             if-state-type
  +++ro hello-timer?       rt-types:
    rtimer-value-seconds16
  +++ro wait-timer?         rt-types:
    rtimer-value-seconds16
++-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 discontinuity-time yang:date-and-time
  +-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
      +-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? rt-types:
        rtimer-value-seconds16
    +-ro statistics
      +-ro discontinuity-time yang:date-and-time
      +-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
2.7. 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 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
        ++rw authentication
          ++rw (auth-type-selection)?
            ++: (ospfv2-auth)
              ++rw ospfv2-auth-trailer-rfc?
                ospfv2-auth-trailer-rfc-version
                {ospfv2-authentication-trailer}?
2.8. OSPF Notifications

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/
+--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/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?       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?
Internet-Draft            OSPF YANG Data Model              October 2019

| +  -> /rt:routing/control-plane-protocols/
| +  control-plane-protocol
| +  [rt:name=current()//..routing-protocol-name]/
| +  ospf:ospf/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/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/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/af
### 2.9. OSPF RPC Operations

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

- **clear-database**: reset the content of a particular OSPF Link State Database.
- **clear-neighbor**: Reset a particular OSPF neighbor or group of neighbors associated with an OSPF interface.

```yang
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

The following RFCs and drafts are not referenced in the document text but are referenced in the ietf-ospf.yang module: [RFC0905], [RFC4576], [RFC4973], [RFC5250], [RFC5309], [RFC5642], [RFC5881], [RFC6991], [RFC7779], [RFC7884], [RFC8294], and [RFC8476].

<CODE BEGINS> file "ietf-ospf@2019-10-17.yang"
module ietf-ospf { yang-version 1.1;"
namespace "urn:ietf:params:xml:ns:yang:ietf-ospf";

prefix ospf;

import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface Management (NMDA Version)";
}

import ietf-routing-types {
    prefix "rt-types";
    reference "RFC 8294: Common YANG Data Types for the Routing Area";
}

import iana-routing-types {
    prefix "iana-rt-types";
    reference "RFC 8294: Common YANG Data Types for the Routing Area";
}

import ietf-routing {
    prefix "rt";
    reference "RFC 8349: A YANG Data Model for Routing Management (NMDA Version)";
}

import ietf-key-chain {
    prefix "key-chain";
    reference "RFC 8177: YANG Data Model for Key Chains";
}

import ietf-bfd-types {
    prefix "bfd-types";
    reference "RFC YYYY: YANG Data Model for Bidirectional Forwarding Detection (BFD). Please replace YYYY with published RFC number for draft-ietf-bfd-yang.";
}
organization
  "IETF LSR - Link State Routing Working Group";

contact
  "WG Web:   <https://datatracker.ietf.org/group/lsr/>
  WG List:  <mailto:lsr@ietf.org>
  Editor:   Derek Yeung
            <mailto:derek@arrcus.com>
  Author:   Acee Lindem
            <mailto:acee@cisco.com>
  Author:   Yingzhen Qu
            <mailto:yingzhen.qu@futurewei.com>
  Author:   Salih K A
            <mailto:salih@juniper.net>
  Author:   Ing-Wher Chen
            <mailto:ingwherchen@mitre.org>";

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.

  This YANG model conforms to the Network Management
  Datastore Architecture (NMDA) as described in RFC 8242.

  Copyright (c) 2018 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

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
  for full legal notices.

  ‘MAY’, and ‘OPTIONAL’ in this document are to be interpreted as

described in BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

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

revision 2019-10-17 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for OSPF.";
}

feature multi-topology {
  description
    "Support Multiple-Topology Routing (MTR).";
  reference "RFC 4915: Multi-Topology Routing";
}

feature multi-area-adj {
  description
    "OSPF multi-area adjacency support as in RFC 5185.";
  reference "RFC 5185: Multi-Area Adjacency";
}

feature explicit-router-id {
  description
    "Set Router-ID per instance explicitly.";
}

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

feature mtu-ignore {
  description
    "Disable OSPF Database Description packet MTU mismatch checking specified in the OSPF protocol specification.";
  reference "RFC 2328: OSPF Version 2, section 10.6";
}

feature lls {
  description
    "OSPF link-local signaling (LLS) as in RFC 5613.";
  reference "RFC 5613: OSPF Link-Local Signaling";
}
feature prefix-suppression {
    description
        "OSPF prefix suppression support as in RFC 6860.";
    reference "RFC 6860: Hide Transit-Only Networks in OSPF";
}

feature ttl-security {
    description
        "OSPF Time to Live (TTL) security check support.";
    reference "RFC 5082: The Generalized TTL Security
        Mechanism (GTSM)";
}

feature nsr {
    description
        "Non-Stop-Routing (NSR) support. The OSPF NSR feature
        allows a router with redundant control-plane capability
        (e.g., dual Route-Processor (RP) cards) to maintain its
        state and adjacencies during planned and unplanned
        OSPF instance restarts. It differs from graceful-restart
        or Non-Stop Forwarding (NSF) in that no protocol signaling
        or assistance from adjacent OSPF neighbors is required to
        recover control-plane state.";
}

feature graceful-restart {
    description
        "Graceful OSPF Restart as defined in RFC 3623 and
        RFC 5187.";
    reference "RFC 3623: Graceful OSPF Restart
        RFC 5187: OSPFv3 Graceful Restart";
}

feature auto-cost {
    description
        "Calculate OSPF interface cost according to
        reference bandwidth.";
    reference "RFC 2328: OSPF Version 2";
}

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

feature max-lsa {
    description
        "Setting the maximum number of LSAs the OSPF instance
        instance.
        
        
feature te-rid {

description
"Support configuration of the Traffic Engineering (TE) Router-ID, i.e., the Router Address described in Section 2.4.1 of RFC3630 or the Router IPv6 Address TLV described in Section 3 of RFC5329.";

reference "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2"
reference "RFC 5329: Traffic Engineering (TE) Extensions to OSPF Version 3";
}

feature ldp-igp-sync {

description
"LDP IGP synchronization.";

reference "RFC 5443: LDP IGP Synchronization";
}

feature ospfv2-authentication-trailer {

description
"Support OSPFv2 authentication trailer for OSPFv2 authentication.";

reference "RFC 5709: Supporting Authentication Trailer for OSPFv2"
}

feature ospfv3-authentication-ipsec {

description
"Support IPsec for OSPFv3 authentication.";

reference "RFC 4552: Authentication/Confidentiality for OSPFv3";
}

feature ospfv3-authentication-trailer {

description
"Support OSPFv3 authentication trailer for OSPFv3 authentication.";

reference "RFC 7166: Supporting Authentication Trailer for OSPFv3";
}

feature fast-reroute {

description
  "Support for IP Fast Reroute (IP-FRR).";
reference "RFC 5714: IP Fast Reroute Framework";
)

feature key-chain {
  description
    "Support of keychain for authentication.";
reference "RFC8177: YANG Data Model for Key Chains";
)

feature node-flag {
  description
    "Support for node-flag for OSPF prefixes.";
reference "RFC 7684: OSPFv2 Prefix/Link Advertisement";
)

feature node-tag {
  description
    "Support for node admin tag for OSPF routing instances.";
reference "RFC 7777: Advertising Node Administrative
Tags in OSPF";
)

feature lfa {
  description
    "Support for Loop-Free Alternates (LFAs).";
reference "RFC 5286: Basic Specification for IP Fast
Reroute: Loop-Free Alternates";
)

feature remote-lfa {
  description
    "Support for Remote Loop-Free Alternates (R-LFA).";
reference "RFC 7490: Remote Loop-Free Alternate (LFA)
Fast Reroute (FRR)";
)

feature stub-router {
  description
    "Support for RFC 6987 OSPF Stub Router Advertisement.";
reference "RFC 6987: OSPF Stub Router Advertisement";
)

feature pe-ce-protocol {
  description
    "Support for OSPF as a PE-CE protocol";
reference "RFC 4577: OSPF as the Provider/Customer Edge
Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)
RFC 6565: OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol;

feature ietf-spf-delay {
    description "Support for IETF SPF delay algorithm.";
    reference "RFC 8405: SPF Back-off algorithm for link state IGPs";
}

feature bfd {
    description "Support for BFD detection of OSPF neighbor reachability.";
    reference "RFC 5880: Bidirectional Forwarding Detection (BFD)
RFC 5881: Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)";
}

feature hybrid-interface {
    description "Support for OSPF Hybrid interface type.";
    reference "RFC 6845: OSPF Hybrid Broadcast and Point-to-Multipoint Interface Type";
}

identity ospf {
    base "rt:routing-protocol";
    description "Any OSPF protocol version";
}

identity ospfv2 {
    base "ospf";
    description "OSPFv2 protocol";
}

identity ospfv3 {
    base "ospf";
    description "OSPFv3 protocol";
}

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

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

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

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

identity nssa-area {
  base stub-nssa-area;
  description "OSPF Not-So-Stubby Area (NSSA).";
  reference "RFC 3101: The OSPF Not-So-Stubby Area (NSSA) Option";
}

identity ospf-lsa-type {
  description
    "Base identity for OSPFv2 and OSPFv3 Link State Advertisement (LSA) types";
}

identity ospfv2-lsa-type {
  base ospf-lsa-type;
  description
    "OSPFv2 LSA types";
}

identity ospfv2-router-lsa {
  base ospfv2-lsa-type;
  description
    "OSPFv2 Router LSA - Type 1";
}

identity ospfv2-network-lsa {
  base ospfv2-lsa-type;
  description
    "OSPFv2 Network LSA - Type 2";
}

identity ospfv2-summary-lsa-type {
  base ospfv2-lsa-type;
  description

"OSPFv2 Summary LSA types";
}

identity ospfv2-network-summary-lsa {
  base ospfv2-summary-lsa-type;
  description
    "OSPFv2 Network Summary LSA - Type 3";
}

identity ospfv2-asbr-summary-lsa {
  base ospfv2-summary-lsa-type;
  description
    "OSPFv2 AS Boundary Router (ASBR) Summary LSA - Type 4";
}

identity ospfv2-external-lsa-type {
  base ospfv2-lsa-type;
  description
    "OSPFv2 External LSA types";
}

identity ospfv2-as-external-lsa {
  base ospfv2-external-lsa-type;
  description
    "OSPFv2 AS External LSA - Type 5";
}

identity ospfv2-nssa-lsa {
  base ospfv2-external-lsa-type;
  description
    "OSPFv2 Not-So-Stubby-Area (NSSA) LSA - Type 7";
}

identity ospfv2-opaque-lsa-type {
  base ospfv2-lsa-type;
  description
    "OSPFv2 Opaque LSA types";
}

identity ospfv2-link-scope-opaque-lsa {
  base ospfv2-opaque-lsa-type;
  description
    "OSPFv2 Link-Scoped Opaque LSA - Type 9";
}

identity ospfv2-area-scope-opaque-lsa {
  base ospfv2-opaque-lsa-type;
  description
"OSPFv2 Area Scoped Opaque LSA - Type 10"; }

identity ospfv2-as-scope-opaque-lsa {
  base ospfv2-opaque-lsa-type;
  description
    "OSPFv2 AS Scoped Opaque LSA - Type 11";
}

identity ospfv2-unknown-lsa-type {
  base ospfv2-lsa-type;
  description
    "OSPFv2 Unknown LSA type";
}

identity ospfv3-lsa-type {
  base ospf-lsa-type;
  description
    "OSPFv3 LSA types.";
}

identity ospfv3-router-lsa {
  base ospfv3-lsa-type;
  description
    "OSPFv3 Router LSA - Type 0x2001";
}

identity ospfv3-network-lsa {
  base ospfv3-lsa-type;
  description
    "OSPFv3 Network LSA - Type 0x2002";
}

identity ospfv3-summary-lsa-type {
  base ospfv3-lsa-type;
  description
    "OSPFv3 Summary LSA types";
}

identity ospfv3-inter-area-prefix-lsa {
  base ospfv3-summary-lsa-type;
  description
    "OSPFv3 Inter-area Prefix LSA - Type 0x2003";
}

identity ospfv3-inter-area-router-lsa {
  base ospfv3-summary-lsa-type;
  description
"OSPFv3 Inter-area Router LSA - Type 0x2004";
}

identity ospfv3-external-lsa-type {
    base ospfv3-lsa-type;
    description
        "OSPFv3 External LSA types";
}

identity ospfv3-as-external-lsa {
    base ospfv3-external-lsa-type;
    description
        "OSPFv3 AS-External LSA - Type 0x4005";
}

identity ospfv3-nssa-lsa {
    base ospfv3-external-lsa-type;
    description
        "OSPFv3 Not-So-Stubby-Area (NSSA) LSA - Type 0x2007";
}

identity ospfv3-link-lsa {
    base ospfv3-lsa-type;
    description
        "OSPFv3 Link LSA - Type 0x0008";
}

identity ospfv3-intra-area-prefix-lsa {
    base ospfv3-lsa-type;
    description
        "OSPFv3 Intra-area Prefix LSA - Type 0x2009";
}

identity ospfv3-router-information-lsa {
    base ospfv3-lsa-type;
    description
        "OSPFv3 Router Information LSA - Types 0x800C, 0xA00C, and 0xC00C";
}

identity ospfv3-unknown-lsa-type {
    base ospfv3-lsa-type;
    description
        "OSPFv3 Unknown LSA type";
}

identity lsa-log-reason {
    description

Yeung, et al. Expires April 19, 2020 [Page 34]
"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.";
}

identity informational-capability {
    description
        "Base identity for router informational capabilities.";
}

identity graceful-restart {
    base informational-capability;
    description
        "When set, the router is capable of restarting
         gracefully.";
    reference "RFC 3623: Graceful OSPF Restart
         RFC 5187: OSPFv3 Graceful Restart";
}

identity graceful-restart-helper {
    base informational-capability;
    description
        "When set, the router is capable of acting as
         a graceful restart helper.";
    reference "RFC 3623: Graceful OSPF Restart
         RFC 5187: OSPFv3 Graceful Restart";
}
identity stub-router {
    base informational-capability;
    description "When set, the router is capable of acting as an OSPF Stub Router.";
    reference "RFC 6987: OSPF Stub Router Advertisement";
}

identity traffic-engineering {
    base informational-capability;
    description "When set, the router is capable of OSPF traffic engineering.";
    reference "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2
                RFC 5329: Traffic Engineering (TE) Extensions to OSPF Version 3";
}

identity p2p-over-lan {
    base informational-capability;
    description "When set, the router is capable of OSPF Point-to-Point over LAN.";
    reference "RFC 5309: Point-to-Point Operation over LAN in Link State Routing Protocols";
}

identity experimental-te {
    base informational-capability;
    description "When set, the router is capable of OSPF experimental traffic engineering.";
    reference "RFC 4973: OSPF-xTE OSPF Experimental Traffic Engineering";
}

identity router-lsa-bit {
    description "Base identity for Router-LSA bits.";
}

identity vlink-end-bit {
    base router-lsa-bit;
    description "V bit, when set, the router is an endpoint of one or more virtual links.";
}
identity asbr-bit {
    base router-lsa-bit;
    description
        "E bit, when set, the router is an AS Boundary Router (ASBR).";
}

identity abr-bit {
    base router-lsa-bit;
    description
        "B bit, when set, the router is an Area Border Router (ABR).";
}

identity nssa-bit {
    base router-lsa-bit;
    description
        "Nt bit, when set, the router is an NSSA border router that is unconditionally translating NSSA LSAs into AS-external LSAs.";
}

identity ospfv3-lsa-option {
    description
        "Base identity for OSPF LSA options flags.";
}

identity af-bit {
    base ospfv3-lsa-option;
    description
        "AF bit, when set, the router supports OSPFv3 Address Families as in RFC5838.";
}

identity dc-bit {
    base ospfv3-lsa-option;
    description
        "DC bit, when set, the router supports demand circuits.";
}

identity r-bit {
    base ospfv3-lsa-option;
    description
        "R bit, when set, the originator is an active router.";
}
identity n-bit {
  base ospfv3-lsa-option;
  description
    "N bit, when set, the router is attached to an NSSA";
}

identity e-bit {
  base ospfv3-lsa-option;
  description
    "E bit, this bit describes the way AS-external LSAs
    are flooded";
}

identity v6-bit {
  base ospfv3-lsa-option;
  description
    "V6 bit, if clear, the router/link should be excluded
    from IPv6 routing calculation";
}

identity ospfv3-prefix-option {
  description
    "Base identity for OSPFv3 Prefix Options.";
}

identity nu-bit {
  base ospfv3-prefix-option;
  description
    "NU Bit, when set, the prefix should be excluded
    from IPv6 unicast calculations.";
}

identity la-bit {
  base ospfv3-prefix-option;
  description
    "LA bit, when set, the prefix is actually an IPv6
    interface address of the Advertising Router.";
}

identity p-bit {
  base ospfv3-prefix-option;
  description
    "P bit, when set, the NSSA area prefix should be
    translated to an AS External LSA and advertised
    by the translating NSSA Border Router.";
}

identity dn-bit {
base ospfv3-prefix-option;
description
  "DN bit, when set, the inter-area-prefix LSA or
  AS-external LSA prefix has been advertised as an
  L3VPN prefix.";
}

identity ospfv2-lsa-option {
  description
  "Base identity for OSPFv2 LSA option flags.";
}

identity mt-bit {
  base ospfv2-lsa-option;
  description
  "MT bit, When set, the router supports multi-topology as
   in RFC 4915.";
}

identity v2-dc-bit {
  base ospfv2-lsa-option;
  description
  "DC bit, When set, the router supports demand circuits.";
}

identity v2-p-bit {
  base ospfv2-lsa-option;
  description
  "P bit, wnlly used in type-7 LSA. When set, an NSSA
   border router should translate the type-7 LSA
   to a type-5 LSA.";
}

identity mc-flag {
  base ospfv2-lsa-option;
  description
  "MC Bit, when set, the router supports MOSPF.";
}

identity v2-e-flag {
  base ospfv2-lsa-option;
  description
  "E Bit, this bit describes the way AS-external LSAs
   are flooded.";
}

identity o-bit {
  base ospfv2-lsa-option;
description
"O bit, when set, the router is opaque-capable as in RFC 5250."
}

identity v2-dn-bit {
  base ospfv2-lsa-option;
  description
  "DN bit, 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."
}

identity ospfv2-extended-prefix-flag {
  description
  "Base identity for extended prefix TLV flag."
}

identity a-flag {
  base ospfv2-extended-prefix-flag;
  description
  "Attach flag, when set it indicates that the prefix corresponds and a route what is directly connected to the advertising router."
}

identity node-flag {
  base ospfv2-extended-prefix-flag;
  description
  "Node flag, when set, it indicates that the prefix is used to represent the advertising node, e.g., a loopback address."
}

typedef ospf-metric {
  type uint32 {
    range "0 .. 16777215"
  }
  description
  "OSPF Metric - 24-bit unsigned integer."
}

typedef ospf-link-metric {
  type uint16 {
    range "0 .. 65535"
  }
  description
  "OSPF Link Metric - 16-bit unsigned integer."
}
typedef opaque-id {
type uint32 {
  range "0 .. 16777215";
} 
description 
"Opaque ID - 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 router-link-type {
  type enumeration {
    enum point-to-point-link {
      value "1";
      description
      "Point-to-Point link to Router";
    }
    enum transit-network-link {
      value "2";
      description
      "Link to transit network identified by Designated-Router (DR)";
    }
    enum stub-network-link {
      value "3";
      description

"Link to stub network identified by subnet";
}
enum virtual-link {
  value "4";
  description
  "Virtual link across transit area";
}
}
description
  "OSPF Router Link 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 "Restart not attempted."
        }
        enum in-progress {
            value "2";
            description "Restart in progress."
        }
        enum completed {
            value "3";
            description "Restart successfully completed."
        }
        enum timed-out {
            value "4";
            description
"Restart timed out."

enum topology-changed {
  value "5"
  description
  "Restart 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-description {
      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 Acknowledgement packet."
    }
  }
  description
  "OSPF packet type."
}

typedef nssa-translator-state-type {
  type enumeration {

enum enabled {
    value "1";
    description "NSSA translator enabled state.";
}
enum elected {
    value "2";
    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 {
            value "2";
            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 fletcher-checksum16-type {
    type string {
        pattern '(0x)?[0-9a-fA-F]{4}';
    }
    description "Fletcher 16-bit checksum in hex-string format 0xXXXX.";
}
typedef ospfv2-auth-trailer-rfc-version {
  type enumeration {
    enum rfc5709 {
      description "Support OSPF Authentication Trailer as described in RFC 5709";
      reference "RFC 5709: OSPFv2 HMAC-SHA Cryptographic Authentication";
    }
    enum rfc7474 {
      description "Support OSPF Authentication Trailer as described in RFC 7474";
    }
  }
  description "OSPFv2 Authentication Trailer Support";
}

grouping tlv {
  description "Type-Length-Value (TLV)";
  leaf type {
    type uint16;
    description "TLV type.";
  }
  leaf length {
    type uint16;
    description "TLV length (octets).";
  }
  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 admin tag value."
        }
        description "List of tags."
    }
}

grouping router-capabilities-tlv {
    description "OSPF Router Capabilities TLV grouping."
    reference "RFC 7770: OSPF Router Capabilities"
    container router-informational-capabilities {
        leaf-list informational-capabilities {
            type identityref {
                base informational-capability;
            }
            description "Informational capability list. This list will contains the identities for the informational capabilities supported by router."
        }
        description "OSPF Router Informational Flag Definitions."
    }
    list informational-capabilities-flags {
        leaf informational-flag {
            type uint32;
            description "Individual informational capability flag."
        }
        description "List of informational capability flags. This will return all the 32-bit informational flags irrespective
of whether or not they are known to the device."
}
list functional-capabilities {
  leaf functional-flag {
    type uint32;
    description "Individual functional capability flag.";
  }
  description "List of functional capability flags. This will return all the 32-bit functional flags irrespective of whether or not they are known to the device.";
}
}

grouping dynamic-hostname-tlv {
  description "Dynamic Hostname TLV";
  reference "RFC 5642: Dynamic Hostnames for OSPF";
  leaf hostname {
    type string {
      length "1..255";
    }
    description "Dynamic Hostname";
  }
}

grouping sbfd-discriminator-tlv {
  description "Seamless BFD Discriminator TLV";
  reference "RFC 7884: S-BFD Discriminators in OSPF";
  list sbfd-discriminators {
    leaf sbfd-discriminator {
      type uint32;
      description "Individual S-BFD Discriminator.";
    }
    description "List of S-BFD Discriminators";
  }
}

grouping maximum-sid-depth-tlv {
  description "Maximum SID Depth (MSD) TLV";
  reference "RFC 8476: Signaling Maximum Segment Depth (MSD) using OSPF";
  list msd-type {
    leaf msd-type {
      type uint8;
      description "Maximum Segment Depth (MSD) type";
    }
  }
}
leaf msd-value {
    type uint8;
    description "Maximum Segment Depth (MSD) value for the type";
}

description "List of Maximum Segment Depth (MSD) tuples";

} 

} 

} 

} 

} 

grouping ospf-router-lsa-bits {
    container router-bits {
        leaf-list rtr-lsa-bits {
            type identityref {
                base router-lsa-bit;
            }
            description "Router LSA bits list. This list will contain identities for the bits which are set in the Router-LSA bits.";
        }
        description "Router LSA Bits.";
    }
    description "Router LSA Bits - Currently common for OSPFv2 and OSPFv3 but it may diverge with future augmentations.";
}

grouping ospfv2-router-link {
    description "OSPFv2 router link.";
    leaf link-id {
        type union {
            type inet:ipv4-address;
            type yang:dotted-quad;
        }
        description "Router-LSA Link ID";
    }
    leaf link-data {
        type union {
            type inet:ipv4-address;
            type uint32;
        }
        description "Router-LSA Link data.";
    }
    leaf type {
        type router-link-type;
        description "Router-LSA Link type.";
    }
}
grouping ospfv2-lsa-body {
  description "OSPFv2 LSA body.";
  container router {
    when "derived-from-or-self(../header/type, " + "'ospfv2-router-lsa')" {
      description
      "Only applies to Router-LSAs.";
    }
    description
    "Router LSA.";
    uses ospf-router-lsa-bits;
    leaf num-of-links {
      type uint16;
      description "Number of links in Router LSA.";
    }
    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 "derived-from-or-self(../header/type, " + "'ospfv2-network-lsa')" {
    description
    "Only applies to Network LSAs.";
  }
}
description
"Network LSA.";
leaf network-mask {
  type yang:dotted-quad;
  description
  "The IP address mask for the network.";
}
container attached-routers {
  description "All attached routers.";
  leaf-list attached-router {
    type inet:ipv4-address;
    description
    "List of the routers attached to the network.";
  }
}
}
}

container summary {
  when "derived-from(../..//header/type, "
   + "'ospfv2-summary-lsa-type')" {
    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 ospf-metric;
        description "Metric for the topology.";
      }
    }
  }
}
container external {
  when "derived-from(../../header/type, " + "'ospfv2-external-lsa-type')"
  { 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 ospf-metric;
    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 "derived-from(../../header/type, " + "'ospfv2-opaque-lsa-type')"
    description "Only applies to Opaque LSAs.";
}
description "Opaque LSA."

container ri-opaque {
  description "OSPF Router Information (RI) opaque LSA.";
  reference "RFC 7770: OSPF Router Capabilities"

  container router-capabilities-tlv {
    description "Informational and functional router capabilities";
    uses router-capabilities-tlv;
  }

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

  container dynamic-hostname-tlv {
    description "OSPF Dynamic Hostname"
    uses dynamic-hostname-tlv;
  }

  container sbfd-discriminator-tlv {
    description "OSPF S-BFD Discriminators"
    uses sbfd-discriminator-tlv;
  }

  container maximum-sid-depth-tlv {
    description "OSPF Maximum SID Depth (MSD) values"
    uses maximum-sid-depth-tlv;
  }
  uses unknown-tlvs;
}

container te-opaque {
  description "OSPFv2 Traffic Engineering (TE) opaque LSA.";
  reference "RFC 3630: Traffic Engineering (TE)
              Extensions to OSPFv2";
}

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

container link-tlv {
  description "Describes a single link, and it is constructed
              of a set of Sub-TLVs.";
  leaf link-type {
    type router-link-type;
    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-opaque {
    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.";
}
container flags {
    leaf-list extended-prefix-flags {
        type identityref {
            base ospfv2-extended-prefix-flag;
        }
        description "Extended prefix TLV flags list. This list will contain identities for the prefix flags that are set in the extended prefix flags.";
    }
    description "Prefix Flags.";
 }
leaf prefix {
    type inet:ip-prefix;
    description "Address prefix.";
}
uses unknown-tlvs;
}
}
container extended-link-opaque {
    description "All extended link TLVs in the LSA.";
    container extended-link-tlv {
        description "Extended link TLV.";
        uses ospfv2-router-link;
        container maximum-sid-depth-tlv {
            description "OSPF Maximum SID Depth (MSD) values";
            uses maximum-sid-depth-tlv;
        }
    }
}
uses unknown-tlvs;
}
}
}

grouping ospfv3-lsa-options {
    description "OSPFv3 LSA options";
    container lsa-options {
        leaf-list lsa-options {
            type identityref {
                base ospfv3-lsa-option;
            }
            description "OSPFv3 LSA Option flags list. This list will contain
            the identities for the OSPFv3 LSA options that are
            set for the LSA.";
        }
        description "OSPFv3 LSA options.";
    }
}

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

    leaf prefix {
        type inet:ip-prefix;
        description "LSA Prefix.";
    }
    container prefix-options {
        leaf-list prefix-options {
            type identityref {
                base ospfv3-prefix-option;
            }
            description "OSPFv3 prefix option flag list. This list will
            contain the identities for the OSPFv3 options
            that are set for the OSPFv3 prefix.";
        }
        description "Prefix options.";
    }
}

grouping ospfv3-lsa-external {
    description "AS-External and NSSA LSA.";
}
leaf metric {
    type ospf-metric;
    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 identityref {
        base ospfv3-lsa-type;
    }
    description "Referenced Link State type.";
}

leaf unknown-referenced-ls-type {
    type uint16;
    description
        "Value for an unknown 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 "derived-from-or-self(../../../header/type, "
      + "'ospfv3-router-lsa')" {
      description
        "Only applies to Router LSAs.";
    }
    description "Router LSA.";
    uses ospf-router-lsa-bits;
    uses ospfv3-lsa-options;
  }
  container links {
    description "All router link.";
    list link {
      description "Router LSA link.";
      leaf interface-id {
        type uint32;
        description "Interface ID for link.";
      }
      leaf neighbor-interface-id {
        type uint32;
        description "Neighbor’s Interface ID for link.";
      }
      leaf neighbor-router-id {
        type rt-types:router-id;
        description "Neighbor’s Router ID for link.";
      }
      leaf type {
        type router-link-type;
        description "Link type: 1 - Point-to-Point Link
          2 - Transit Network Link
          3 - Stub Network Link
          4 - Virtual Link";
      }
      leaf metric {
        type uint16;
        description "Link Metric.";
      }
    }
  }
}
container network {
    when "derived-from-or-self(../header/type, " + "+" ospfv3-network-lsa")" {
        description
            "Only applies to Network LSAs.";
    }
    description "Network LSA."
    uses ospfv3-lsa-options;
    container attached-routers {
        description "All attached routers.";
        leaf-list attached-router {
            type rt-types:router-id;
            description
                "List of the routers attached to the network.";
        }
    }
}

container inter-area-prefix {
    when "derived-from-or-self(../header/type, " + "+" ospfv3-inter-area-prefix-lsa")" {
        description
            "Only applies to Inter-Area-Prefix LSAs.";
    }
    leaf metric {
        type ospf-metric;
        description "Inter-Area Prefix Metric";
    }
    uses ospfv3-lsa-prefix;
    description "Prefix LSA.";
}

container inter-area-router {
    when "derived-from-or-self(../header/type, " + "+" ospfv3-inter-area-router-lsa")" {
        description
            "Only applies to Inter-Area-Router LSAs.";
    }
    uses ospfv3-lsa-options;
    leaf metric {
        type ospf-metric;
        description "AS Boundary Router (ASBR) Metric.";
    }
    leaf destination-router-id {
        type rt-types:router-id;
    }
}
description "The Router ID of the ASBR described by the LSA."
}

container as-external {
  when "derived-from-or-self(../../header/type, " + "'ospfv3-as-external-lsa')" {
    description "Only applies to AS-external LSAs."
  }
  uses ospfv3-lsa-external;
  description "AS-External LSA."
}

container nssa {
  when "derived-from-or-self(../../header/type, " + "'ospfv3-nssa-lsa')" {
    description "Only applies to NSSA LSAs."
  }
  uses ospfv3-lsa-external;
  description "NSSA LSA."
}

container link {
  when "derived-from-or-self(../../header/type, " + "'ospfv3-link-lsa')" {
    description "Only applies to Link LSAs."
  }
  leaf rtr-priority {
    type uint8;
    description "Router priority for DR election. A router with a higher priority will be preferred in the election and a value of 0 indicates the router is not eligible to become Designated Router or Backup Designated Router (BDR)."
  }
  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 "derived-from-or-self(../../../header/type, " + "'ospfv3-intra-area-prefix-lsa')" {
    description "Only applies to Intra-Area-Prefix LSAs.";
  }
}

description "Intra-Area-Prefix LSA."

leaf referenced-ls-type {
  type identityref {
    base ospfv3-lsa-type;
  }
  description "Referenced Link State type."
}

leaf unknown-referenced-ls-type {
  type uint16;
  description "Value for an unknown Referenced Link State type."
}

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

leaf referenced-adv-router {
  type rt-types:router-id;
  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 ospf-metric;
            description "Prefix Metric.";
        }
    }
}

container router-information {
    when "derived-from-or-self(../../header/type, "
        + "'ospfv3-router-information-lsa')" {
        description "Only applies to Router Information LSAs (RFC7770).";
    }
    container router-capabilities-tlv {
        description "Informational and functional router capabilities";
        uses router-capabilities-tlv;
    }
    container node-tag-tlvs {
        description "All node tag tlvs.";
        list node-tag-tlv {
            description "Node tag tlv.";
            uses node-tag-tlv;
        }
    }
    container dynamic-hostname-tlv {
        description "OSPF Dynamic Hostname";
        uses dynamic-hostname-tlv;
    }
    container sbfd-discriminator-tlv {
        description "OSPF S-BFD Discriminators";
        uses sbfd-discriminator-tlv;
    }
    description "Router Information LSA.";
    reference "RFC 7770: Extensions for Advertising Router Capabilities";
}
grouping lsa-header {
    description "Common LSA for OSPFv2 and OSPFv3";
    leaf age {
        type uint16;
        mandatory true;
        description "LSA age.";
    }
    leaf type {
        type identityref {
            base ospf-lsa-type;
        }
        mandatory true;
        description "LSA type";
    }
    leaf adv-router {
        type rt-types:router-id;
        mandatory true;
        description "LSA advertising router.";
    }
    leaf seq-num {
        type uint32;
        mandatory true;
        description "LSA sequence number.";
    }
    leaf checksum {
        type fletcher-checksum16-type;
        mandatory true;
        description "LSA checksum.";
    }
    leaf length {
        type uint16;
        mandatory true;
        description "LSA length including the header.";
    }
}

grouping ospfv2-lsa {
    description "OSPFv2 LSA - LSAs are uniquely identified by the <LSA Type, Link-State ID, Advertising Router> tuple with the sequence number differentiating LSA instances.";
    container header {
        must "(derived-from(type, " + "'ospfv2-opaque-lsa-type') and " + "opaque-id and opaque-type) or " + "(not(derived-from(type, " + "'ospfv2-opaque-lsa-type') and " + "opaque-type)) and " + "opaque-id)";
    }
}
+ "'ospfv2-opaque-lsa-type')" "
+ "and not(opaque-id) and not(opaque-type))" { 
  description
  "Opaque type and ID only apply to Opaque LSAs.";
}
  description
  "Decoded OSPFv2 LSA header data.";
container lsa-options {
  leaf-list lsa-options {
    type identityref {
      base ospfv2-lsa-option;
    }
    description
    "LSA option flags list. This list will contain
    the identities for the OSPFv2 LSA options that are set.";
  }
  description
  "LSA options.";
}
leaf lsa-id {
  type yang:dotted-quad;
  mandatory true;
  description "Link-State ID.";
}
leaf opaque-type {
  type uint8;
  description "Opaque type.";
}
leaf opaque-id {
  type opaque-id;
  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 "OSPFv3 LSA ID.";
  }
  uses lsa-header;
}
container body {
  description
    "Decoded OSPF LSA body data.";
  uses ospfv3-lsa-body;
}
}
grouping lsa-common {
  description
    "Common fields for OSPF LSA representation.";
  leaf decode-completed {
    type boolean;
    description
      "The OSPF LSA body was successfully decoded other than
      unknown TLVs. Unknown LSAs types and OSPFv2 unknown
      opaque LSA types are not decoded. Additionally,
      malformed LSAs are generally not accepted and will
      not be in the Link State Database.";
  }
  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;
}

}
type yang:counter32;
description
"The number of new LSAs received. Discontinuities in the
value of this counter can occur when the OSPF instance is
re-initialized."
}
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 module 2**32 sum of the LSA checksums
for AS-scope LSAs. The value should be treated as
unsigned when comparing two sums of checksums. While
differing checksums indicate a different combination
of LSAs, equivalent checksums don't guarantee that the
LSAs are the same given that multiple combinations of
LSAs can result in the same checksum."
}
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 uint32;
description
"The module 2**32 sum of the LSA checksums
for the LSAs of this type. The value should be
treated as unsigned when comparing two sums of
checksums. While differing checksums indicate a
different combination of LSAs, equivalent checksums
don't guarantee that the LSAs are the same given that
multiple combinations of LSAs can result in the same
checksum."
    }
  }
}
uses instance-fast-reroute-state;
grouping area-stat {
    description "Per-area statistics.";
    leaf discontinuity-time {
        type yang:date-and-time;
        description "The time on the most recent occasion at which any one or
        more of this OSPF area’s counters suffered a discontinuity. If no such discontinuities have occurred
        since the OSPF area was last re-initialized, then
        this node contains the time the OSPF area was
        re-initialized which normally occurs when it was
        created.";
    }
    leaf spf-runs-count {
        type yang:counter32;
        description "The number of times the intra-area SPF has run.
        Discontinuities in the value of this counter can occur
        when the OSPF area is re-initialized.";
    }
    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 Boundary Routers (ASBRs).";
    }
    leaf ar-nssa-translator-event-count {
        type yang:counter32;
        description "The number of NSSA translator-state changes.
        Discontinuities in the value of this counter can occur
        when the OSPF area is re-initialized.";
    }
    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 uint32;
        description "";
    }
}
"The module $2^{32}$ sum of the LSA checksums for area-scope LSAs. The value should be treated as unsigned when comparing two sums of checksums. While differing checksums indicate a different combination of LSAs, equivalent checksums don't guarantee that the LSAs are the same given that multiple combinations of LSAs can result in the same checksum."

```yang
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 uint32;
      description "The module $2^{32}$ sum of the LSA checksums for the LSAs of this type. The value should be treated as unsigned when comparing two sums of checksums. While differing checksums indicate a different combination of LSAs, equivalent checksums don't guarantee that the LSAs are the same given that multiple combinations of LSAs can result in the same checksum.";
    }
  }

  grouping interface-stat {
    description "Per-interface statistics";
    leaf discontinuity-time {
      type yang:date-and-time;
      description "The time on the most recent occasion at which any one or more of this OSPF interface’s counters suffered a discontinuity. If no such discontinuities have occurred since the OSPF interface was last re-initialized, then this node contains the time the OSPF interface was re-initialized which normally occurs when it was created.";
    }
  }
}
```
leaf if-event-count {
  type yang:counter32;
  description "The number of times this interface has changed its state or an error has occurred. Discontinuities in the value of this counter can occur when the OSPF interface is re-initialized.";
}

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 module 2**32 sum of the LSA checksums for link-scope LSAs. The value should be treated as unsigned when comparing two sums of checksums. While differing checksums indicate a different combination of LSAs, equivalent checksums don’t guarantee that the LSAs are the same given that multiple combinations of LSAs can result in the same checksum.";
}

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 uint32;
      description "The module 2**32 sum of the LSA checksums for the LSAs of this type. The value should be treated as unsigned when comparing two sums of checksums. While differing checksums indicate a different combination of LSAs, equivalent checksums don’t guarantee that the LSAs are the same given that multiple combinations of LSAs can result in the same checksum.";
    }
  }
}
grouping neighbor-stat {
    description "Per-neighbor statistics.";
    leaf discontinuity-time {
        type yang:date-and-time;
        description "The time on the most recent occasion at which any one or more of this OSPF neighbor’s counters suffered a discontinuity. If no such discontinuities have occurred since the OSPF neighbor was last re-initialized, then this node contains the time the OSPF neighbor was re-initialized which normally occurs when the neighbor is dynamically discovered and created.";
    }
    leaf nbr-event-count {
        type yang:counter32;
        description "The number of times this neighbor has changed state or an error has occurred. Discontinuities in the value of this counter can occur when the OSPF neighbor is re-initialized.";
    }
    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 Fast ReRoute (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 Loop-Free Alternatives (LFA). Container creation has no effect on LFA activation.";
    }
}
grouping instance-fast-reroute-state {
  description "IP-FRR state data grouping";

  container protected-routes {
    if-feature fast-reroute;
    config false;
    description "Instance protection statistics";

    list address-family-stats {
      key "address-family prefix alternate";
      description "Per Address Family protected prefix information";

      leaf address-family {
        type iana-rt-types:address-family;
        description "Address-family";
      }

      leaf prefix {
        type inet:ip-prefix;
        description "Protected prefix.";
      }

      leaf alternate {
        type inet:ip-address;
        description "Alternate next hop for the prefix.";
      }

      leaf alternate-type {
        type enumeration {
          enum equal-cost {
            description "ECMP alternate.";
          }

          enum lfa {
            description "LFA alternate.";
          }

          enum remote-lfa {
            description "Remote LFA alternate.";
          }

          enum tunnel {
            description "Tunnel based alternate";
          }
        }
      }
    }
  }
}
enum ti-lfa {
    description "TI-LFA alternate.";
}
enum mrt {
    description "MRT alternate.";
}
enum other {
    description "Unknown alternate type.";
}

description "Type of alternate.";

leaf best {
    type boolean;
    description "Indicates that this alternate is preferred.";
}

leaf non-best-reason {
    type string {
        length "1..255";
    }
    description "Information field to describe why the alternate is not best.";
}

leaf protection-available {
    type bits {
        bit node-protect {
            position 0;
            description "Node protection available.";
        }
        bit link-protect {
            position 1;
            description "Link protection available.";
        }
        bit srlg-protect {
            position 2;
            description "SRLG protection available.";
        }
    }
}
bit downstream-protect {
    position 3;
    description
    "Downstream protection available.";
}

bit other {
    position 4;
    description
    "Other protection available.";
}

description "Protection provided by the alternate.";

leaf alternate-metric1 {
    type uint32;
    description
    "Metric from Point of Local Repair (PLR) to
destination through the alternate path.";
}

leaf alternate-metric2 {
    type uint32;
    description
    "Metric from PLR to the alternate node";
}

leaf alternate-metric3 {
    type uint32;
    description
    "Metric from alternate node to the destination";
}

container unprotected-routes {
    if-feature fast-reroute;
    config false;
    description "List of prefixes that are not protected";

    list address-family-stats {
        key "address-family prefix";
        description
        "Per Address Family (AF) unprotected prefix statistics.";

        leaf address-family {
            type iana-rt-types:address-family;
            description "Address-family";
        }

        leaf prefix {
            type inet:ip-prefix;
        }
    }
}
description "Unprotected prefix."
}
}

list protection-statistics {
  key frr-protection-method;
  config false;
  description "List protection method statistics";

  leaf frr-protection-method {
    type string;
    description "Protection method used.";
  }
}

list address-family-stats {
  key address-family;
  description "Per Address Family protection statistics."

  leaf address-family {
    type iana-rt-types:address-family;
    description "Address-family";
  }

  leaf total-routes {
    type uint32;
    description "Total prefixes.";
  }

  leaf unprotected-routes {
    type uint32;
    description "Total prefixes that are not protected.";
  }

  leaf protected-routes {
    type uint32;
    description "Total prefixes that are protected.";
  }

  leaf linkprotected-routes {
    type uint32;
    description "Total prefixes that are link protected.";
  }

  leaf nodeprotected-routes {
    type uint32;
    description "Total prefixes that are node protected.";
  }
}
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
        "Interface IP Fast-reroute configuration.";
}

grouping interface-physical-link-config {
    description
        "Interface cost configuration that only applies to
         physical interfaces (non-virtual) and sham links.";
    leaf cost {
        type ospf-link-metric;
        description
            "Interface cost configuration that only applies to
             physical interfaces (non-virtual) and sham links.";
    }
}
"Interface cost."
}
leaf mtu-ignore {
  if-feature mtu-ignore;
  type boolean;
  description
      "Enable/Disable bypassing the MTU mismatch check in
       Database Description packets specified in RFC 2328,
       section 10.6.";
}
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;
    units seconds;
    description
      "Interval between hello packets (seconds). It must
       be the same for all routers on the same network.
       Different networks, implementations, and deployments
       will use different hello-intervals. A sample value
       for a LAN network would be 10 seconds.";
    reference "RFC 2328: OSPF Version 2, Appendix C.3";
  }
  leaf dead-interval {
    type uint16;
    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 the ‘hello-interval’.";
    }
    description
      "Interval after which a neighbor is declared down
       (seconds) if hello packets are not received. It is
typically 3 or 4 times the hello-interval. A typical value for LAN networks is 40 seconds.
reference "RFC 2328: OSPF Version 2, Appendix C.3";
}

leaf retransmit-interval {
  type uint16 {
    range "1..3600";
  }
  units seconds;
  description
  "Interval between retransmitting unacknowledged Link State Advertisements (LSAs) (seconds). This should be well over the round-trip transmit delay for any two routers on the network. A sample value would be 5 seconds.";
  reference "RFC 2328: OSPF Version 2, Appendix C.3";
}

leaf transmit-delay {
  type uint16;
  units seconds;
  description
  "Estimated time needed to transmit Link State Update (LSU) packets on the interface (seconds). LSAs have their age incremented by this amount when advertised on the interface. A sample value would be 1 second.";
  reference "RFC 2328: OSPF Version 2, Appendix C.3";
}

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

container ttl-security {
  if-feature ttl-security;
  description "Time to Live (TTL) security check.";
  leaf enable {
    type boolean;
    description
    "Enable/Disable TTL security check.";
  }
  leaf hops {
    type uint8 {
      range "1..254";
    }
  }
}
default 1;

description
"Maximum number of hops that an OSPF packet may have traversed before reception."
);

leaf enable {

type boolean;

default true;

description
"Enable/disable OSPF protocol on the interface."
);

container authentication {

description "Authentication configuration."
;

choice auth-type-selection {

description
"Options for OSPFv2/OSPFv3 authentication configuration."
;

case ospfv2-auth {

when "derived-from-or-self(../../../..//rt:type, " + "'ospfv2')" {

description "Applied to OSPFv2 only.";

leaf ospfv2-auth-trailer-rfc {

if-feature ospfv2-authentication-trailer;

type ospfv2-auth-trailer-rfc-version;

description
"Version of OSPFv2 authentication trailer support - RFC 5709 or RFC 7474";
}

choice ospfv2-auth-specification {

description
"Key chain or explicit key parameter specification";

case auth-key-chain {

if-feature key-chain;

leaf ospfv2-key-chain {


type key-chain:key-chain-ref;

description
"key-chain name.";

}

case auth-key-explicit {

leaf ospfv2-key-id {


type uint32;

description
"Key Identifier";

}
leaf ospfv2-key {
  type string;
  description
  "OSPFv2 authentication key. The
   length of the key may be dependent on the
   cryptographic algorithm.";
}
leaf ospfv2-crypto-algorithm {
  type identityref {
    base key-chain:crypto-algorithm;
  }
  description
  "Cryptographic algorithm associated with key.";
}

case ospfv3-auth-ipsec {
  when "derived-from-or-self(../../../../../../rt:type, " + "+"'ospfv3')" {
    description "Applied to OSPFv3 only.";
  }
  if-feature ospfv3-authentication-ipsec;
  leaf sa {
    type string;
    description
    "Security Association (SA) name.";
  }
}

case ospfv3-auth-trailer {
  when "derived-from-or-self(../../../../../../rt:type, " + "+"'ospfv3')" {
    description "Applied to OSPFv3 only.";
  }
  if-feature ospfv3-authentication-trailer;
  choice ospfv3-auth-specification {
    description
    "Key chain or explicit key parameter specification";
    case auth-key-chain {
      if-feature key-chain;
      leaf ospfv3-key-chain {
        type key-chain:key-chain-ref;
        description
        "key-chain name.";
      }
    }
  };
  case auth-key-explicit {
  }
}
leaf ospfv3-sa-id {
    type uint16;
    description
        "Security Association (SA) Identifier";
}
leaf ospfv3-key {
    type string;
    description
        "OSPFv3 authentication key. The 
        length of the key may be dependent on the 
        cryptographic algorithm.";
}
leaf ospfv3-crypto-algorithm {
    type identityref {
        base key-chain:crypto-algorithm;
    }
    description
        "Cryptographic algorithm associated with key.";
}
}
}
}
}
}


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.";
            }
        }
    }
}

enum "hybrid" {
  if-feature hybrid-interface;
  description "Specify OSPF hybrid broadcast/P2MP network.";
}

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. On multi-access network this value is for Designated Router (DR) election. The priority is ignored on other interface types. A router with a higher priority will be preferred in the election and a value of 0 indicates the router is not eligible to become Designated Router or Backup Designated Router (BDR).";
}

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 ospf-link-metric;
  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 ospf-link-metric;
      description
        "Neighbor cost. Different implementations have different
default costs with some defaulting to a cost inversely
proportional to the interface speed. Others will
default to 1 equating the cost to a hop count."
    }

    leaf poll-interval {
      type uint16;
      units seconds;
      description
        "Neighbor poll interval (seconds) for sending OSPF
hello packets to discover the neighbor on NBMA
networks. This interval dictates the granularity for
discovery of new neighbors. A sample would be
120 seconds (2 minutes) for a legacy Packet Data
Network (PDN) X.25 network.";
      reference "RFC 2328: OSPF Version 2, Appendix C.5"
    }

    leaf priority {
      type uint8;
      description
        "Neighbor priority for DR election. A router with a
higher priority will be preferred in the election"
and a value of 0 indicates the router is not eligible to become Designated Router or Backup Designated Router (BDR)."

```
}
}

leaf node-flag {
  if-feature node-flag;
  type boolean;
  default false;
  description
    "Set prefix as identifying the advertising router.";
  reference "RFC 7684: OSPFv2 Prefix/Link Attribute Advertisement";
}

container bfd {
  if-feature bfd;
  description "BFD Client Configuration.";
  uses bfd-types:client-cfg-parms;
  reference "RFC YYYY: YANG Data Model for Bidirectional Forwarding Detection (BFD). Please replace YYYY with published RFC number for draft-ietf-bfd-yang.";
}

uses interface-fast-reroute-config;
uses interface-common-config;
uses interface-physical-link-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 cost {
  type ospf-link-metric;
  config false;
  description "Cost to reach neighbor for Point-to-Multipoint
    and Hybrid networks";
}

leaf dead-timer {
  type rt-types:timer-value-seconds16;
  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: OSPF Version2 -
    The Interface Data Structure";
}
leaf state {
    type if-state-type;
    config false;
    description "Interface state.";
}

leaf hello-timer {
    type rt-types:timer-value-seconds16;
    config false;
    description "This timer tracks the remaining time before
    the next hello packet is sent on the
    interface."
}

leaf wait-timer {
    type rt-types:timer-value-seconds16;
    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 interface OSPF neighbors.";
    leaf neighbor-router-id {
      type rt-types:router-id;
      description "Neighbor Router ID.";
    }
    uses neighbor-state;
  }
}

container database {
  config false;
  description "Link-scope Link State Database.";
  list link-scope-lsa-type {
    key "lsa-type";
    description "List OSPF link-scope LSAs.";
    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 "derived-from-or-self( "
          + "/.../.../.../.../.../.../.../.../.../"
          + "rt:type, 'ospfv2')" {
            description "OSPFv2 LSA.";
          }
        }
        refine "version/ospfv3/ospfv3" {
          must "derived-from-or-self( "
          + "/.../.../.../.../.../.../.../.../.../"
          + "rt:type, 'ospfv3')" {
          }
        }
      }
    }
  }
}
grouping interface-state {
  description
    "OSPF interface operational state.";
  reference "RFC2328 Section 9: OSPF Version2 -
      The Interface Data Structure";
  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 ospf-link-metric;
    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 {
grouping address-family-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 -
    Applicable to 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 ospf-metric;
      description
      "Advertised cost of summary route.";
    }
  }
}

}
NSSA area.

leaf default-cost {
  when "derived-from(../area-type,'stub-nssa-area')" {
    description
    "Cost for LSA default route advertised into the
     stub or NSSA area."
  }
  type ospf-metric;
  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-area;
    description
    "Area type.";
  }
  uses area-common-config;
  uses address-family-area-config;
}

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 Link State Database.";
    list area-scope-lsa-type {
      key "lsa-type";
      description "List OSPF area-scope LSAs.";
    }
  }
}
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 "derived-from-or-self( "
                + ".../.../.../.../.../..."
                + "rt:type, 'ospfv2')" {
                    description "OSPFv2 LSA.";
                }
            }
            refine "version/ospfv3/ospfv3" {
                must "derived-from-or-self( "
                + ".../.../.../.../.../..."
                + "rt:type, 'ospfv3')" {
                    description "OSPFv3 LSA.";
                }
            }
        }
    }
}

grouping local-rib {
    description "Local-rib - RIB for Routes computed by the local OSPF routing instance.";
    container local-rib {
        config false;
        description "Local-rib.";
        list route {
            key "prefix";
            description "Routes";
            leaf prefix {
                type inet:ip-prefix;
                description "Destination prefix.";
            }
        }
    }
}
list next-hop {
  key "next-hop";
  description "List of next hops for the route";
  leaf outgoing-interface {
    type if:interface-ref;
    description "Name of the outgoing interface.";
  }
  leaf next-hop {
    type inet:ip-address;
    description "Next hop 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 {
  leaf initial-delay {
    type uint32;
    units milliseconds;
    description "Delay used while in QUIET state (milliseconds).";
  }
  leaf short-delay {
    type uint32;
    units milliseconds;
    description "Delay used while in SHORT_WAIT state (milliseconds).";
  }
  leaf long-delay {
    type uint32;
    units milliseconds;
    description
leaf hold-down {
  type uint32;
  units milliseconds;
  description
  "Timer used to consider an IGP stability period
   (milliseconds).";
}

leaf time-to-learn {
  type uint32;
  units milliseconds;
  description
  "Duration used to learn all the IGP events
   related to a single component failure (milliseconds).";
}

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 SPF back-off algorithm state.";
}

leaf remaining-time-to-learn {
  type rt-types:timer-value-milliseconds;
  config false;
  description
  "Remaining time until time-to-learn timer fires.";
}

leaf remaining-hold-down {
  type rt-types:timer-value-milliseconds;
  config false;
  description
  "Remaining time until hold-down timer fires.";
}

leaf last-event-received {
  type yang:timestamp;
  config false;
  description
  "Delay used while in LONG_WAIT state (milliseconds).";
}
"Time of last SPF triggering event."
}
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 configuration and state";
}

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 admin tags.";
  }
}

grouping instance-config {
  description
    "OSPF instance config state.";

  leaf enable {
    type boolean;
    default true;
    description
      "Enable/Disable the protocol.";
  }
}
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 configuration. In many
     implementations, preference is referred to as
     administrative distance.";
  reference
    "RFC 8349: A YANG Data Model for Routing Management
     (NMDA Version)";
  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 intra-area and inter-area routes.";
      }
    }
    }
  }
}
description
    "Preference for both intra-area and
    inter-area routes.";
}
}
leaf external {
  type uint8;
  description
    "Preference for AS 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.";
  reference "RFC 3623: OSPF Graceful Restart
RFC 5187: OSPFv3 Graceful Restart";
  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";
    }
  }
}
units seconds;
default "120";
description
"Interval to attempt graceful restart prior to failing (RFC 3623 Section B.1) (seconds)";
}
leaf helper-strict-lsa-checking {
type boolean;
description
"Terminate graceful restart when an LSA topology change is detected (RFC 3623 Section B.2).";
}

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..65535";
}
description
"Maximum number of Equal-Cost Multi-Path (ECMP) paths."
}
container ietf-spf-delay {
    if-feature ietf-spf-delay;
    uses ietf-spf-delay;
    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 MaxLinkMetric"
            reference "RFC 6987: OSPF Stub Router
                 Advertisement";
        }
    }
}

container mpls {
    description
        "OSPF MPLS config state.";
    container te-rid {
        if-feature te-rid;
        description
            "Stable OSPF Router IP Address used for Traffic
             Youting, et al. Expires April 19, 2020 [Page 100]"
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 database {
config false;
description "AS-scope Link State Database.";
list as-scope-lsa-type {
    key "lsa-type";
description "List OSPF AS-scope LSAs.";
leaf lsa-type {
    type uint16;
description "OSPF AS scope LSA type.";
}
}
container as-scope-lsas {
    description "List OSPF AS-scope LSAs of this LSA type.";
list as-scope-lsa {
    key "lsa-id adv-router";
description "List of OSPF AS-scope LSAs";
uses lsa-key;
uses lsa {
    refine "version/ospfv2/ospfv2" {
        must "derived-from-or-self( "
        + ".../../../../../" 
        + "rt:type, 'ospfv2')" {
            description "OSPFv2 LSA.";
        }
    }
    refine "version/ospfv3/ospfv3" {
        must "derived-from-or-self( "
        + ".../../../../../" 
        + "rt:type, 'ospfv3')" {
            description "OSPFv3 LSA.";
        }
    }
    }
}
}
uses spf-log;
uses lsa-log;
}

grouping multi-topology-area-common-config {
    description "OSPF multi-topology area common configuration state.";
leaf summary {
    when "derived-from(..../..../area-type, 'stub-nssa-area')" {
        description "Summary advertisement into the stub/NSSA area.";
    }
    type boolean;
}
leaf default-cost {
    when "derived-from(../../../area-type, 'stub-nssa-area')" {
        description
            "Cost for LSA default route advertised into the
topology into the stub or NSSA area."
    }
    type ospf-metric;
    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 address-family-area-config;
}

grouping multi-topology-state {
    description
        "OSPF multi-topology operational state.";
    uses local-rib;
}

grouping multi-topology-interface-config {
    description
        "OSPF multi-topology configuration state.";
    leaf cost {
        type ospf-link-metric;
        description
            "Interface cost for this topology."
    }
}

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 type {
    type uint16;
    description
      "LSA type.";
  }

  leaf lsa-id {
    type union {
      type inet:ipv4-address;
      type yang:dotted-quad;
    }
    description "Link-State ID.";
  }

  leaf adv-router {
    type rt-types:router-id;
    description
      "LSA advertising router.";
  }

  leaf seq-num {
    type uint32;
    description
      "Sequence number.";
  }
}
"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 log entries represented
        as a wrapping buffer in chronological
        order with the oldest entry returned
        first.”;
      leaf id {
        type uint32;
        description
          "Event identifier - Purely internal value.”;
      }
      leaf spf-type {
        type enumeration {
          enum full {
            description
              "SPF computation was a Full SPF.”;
          }
          enum intra {
            description
              "SPF computation was only for intra-area routes.”;
          }
          enum inter {
            description
              "SPF computation was only for inter-area
              summary routes.”;
          }
          enum external {
            description
              "SPF computation was only for AS external routes.”;
          }
        }
      }
    description
      "The SPF computation type for the SPF log entry.”;
  }
  leaf schedule-timestamp {
    type yang:timestamp;
leaf start-timestamp {
  type yang:timestamp;
  description
    "This is the timestamp when the computation was started.";
}

leaf end-timestamp {
  type yang:timestamp;
  description
    "This is 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 the LSA log.";
  container lsa-log {
    config false;
    description
      "This container lists the LSA log. Local LSA modifications are also included in the list.";
    list event {
      key id;
      description
        "List of LSA log entries represented as a wrapping buffer in chronological order with the oldest entries returned first.";
      leaf id {
        type uint32;
        description
          "Event identifier - purely internal value.";
      }
      container lsa {
        description
          "This container describes the logged LSA.";
      }
    }
  }
}

description
  "This is the timestamp when the computation was scheduled.";
}
uses lsa-identifiers;
}
leaf received-timestamp {
  type yang:timestamp;
  description
  "This is the timestamp when the LSA was received.
   In case of local LSA update, the timestamp refers
   to the LSA origination time."
}
leaf reason {
  type identityref {
    base lsa-log-reason;
  }
  description
  "This reason for the LSA log entry."
}

augment "/rt:routing/rt:control-plane-protocols/
 + "rt:control-plane-protocol" {
  when "derived-from(rt:type, 'ospf')" {
    description
    "This augmentation is only valid for a routing protocol
     instance of OSPF (type 'ospfv2' or 'ospfv3').";
  }
  description "OSPF protocol ietf-routing module
   control-plane-protocol augmentation.";
}

container ospf {
  description
  "OSPF protocol Instance";

  leaf address-family {
    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 "derived-from-or-self(../area-type, 'normal-area') " + "and ../area-id = '0.0.0.0'" {
        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 "derived-from-or-self(" + "/area[type=normal-area]/area-id=current() = " + "/area[type=normal-area]/area-id != " + "'0.0.0.0'"
                error-message "Virtual link transit area must " + "be non-zero.";
        }
        description
            "Virtual-link transit area must be non-zero area.";
    }
    description
        "Virtual link transit 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 reference.";
    }
    uses interface-config;
    uses interface-state;
  }
}

augment "/rt:routing/rt:control-plane-protocols/" + "rt:control-plane-protocol/ospf" {
  when "derived-from(.../rt:type, 'ospf')" {
    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 {
    key "name"
    description "OSPF topology - The OSPF topology address-family must coincide with the routing-instance address-family."
    leaf name {
      type leafref {
        path "../../../../../../rt:ribs/rt:rib/rt:name"
      }
      description "RIB name corresponding to the OSPF topology."
    }
    uses multi-topology-state;
  }
}

augment "/rt:routing/rt:control-plane-protocols/
  + "rt:control-plane-protocol/ospf/
  + "areas/area"
  when "derived-from-or-self(../../../rt:type, 
  + "/'ospfv2')" {
    description "This augmentation is only valid for OSPFv2."
  }
if-feature multi-topology;
description "OSPF multi-topology area configuration state augmentation."
container topologies {
  description "All topologies for the area."
  list topology {
    key "name"
    description "OSPF area topology."
    leaf name {
      type leafref {
        path "../../../../../../../../rt:ribs/rt:rib/rt:name"
      }
    }
  }
}
description
"Single topology enabled for this area."
}

uses multi-topology-area-config;
}
}

augment "/rt:routing/rt:control-plane-protocols/
+ "rt:control-plane-protocol/ospf/
+ "areas/area/interfaces/interface" {
when "derived-from-or-self(../../../rt:type,
+ "ospfv2")" {
  description
  "This augmentation is only valid for OSPFv2."
}
if-feature 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
      "Single topology enabled on this interface."
    }
    uses multi-topology-interface-config;
  }
}

augment "/rt:routing/rt:control-plane-protocols/
+ "rt:control-plane-protocol/ospf/
+ "areas/area/interfaces/interface" {
when "derived-from-or-self(../../../rt:type,
+ "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 "derived-from(rt:source-protocol, 'ospf')" {
    description
    "This augmentation is only valid for 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
"OSPF protocol instance which information for neighbors
are to be cleared.

If the referenced OSPF instance 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 for which neighbors are to
be cleared.

If the referenced OSPF interface 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 Link State
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
"OSPF protocol instance whose Link State Database is to
be cleared.

If the referenced OSPF instance 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 grouping describes common instance specific
data for OSPF notifications."

  leaf routing-protocol-name {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/
        + "rt:control-plane-protocol/rt:name";
    }
    must "derived-from(" + "/rt:routing/rt:control-plane-protocols/
        + "rt:control-plane-protocol[rt:name=current()]/"
        + "rt:type, 'ospf')";
    description
      "OSPF routing protocol instance name.";
  }

  leaf address-family {
    type leafref {
      path "/rt:routing/
        + "rt:control-plane-protocols/rt:control-plane-protocol"
        + "[rt:name=current()]/../routing-protocol-name"/
        + "ospf/address-family";
    }
    description
      "Address family of the OSPF instance.";
  }
}

grouping notification-interface {
  description
    "This grouping provides interface information
     for the OSPF interface specific notification.";

  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 neighbor specific notifications.";
  leaf neighbor-router-id {
    type rt-types:router-id;
    description "Neighbor Router ID.";
  }
  leaf neighbor-ip-addr {
    type inet:ip-address;
  }
}
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 inet:ip-address;
    description "Source address.";
  }

  leaf packet-type {
    type packet-type;
    description "OSPF packet type.";
  }

  leaf error {
    type enumeration {
      enum "bad-version" {
        description "Bad version.";
      }
      enum "area-mismatch" {
        description "Area mismatch.";
      }
      enum "unknown-nbma-nbr" {
        description "Unknown NBMA neighbor.";
      }
      enum "unknown-virtual-nbr" {
        description "Unknown virtual link neighbor.";
      }
      enum "auth-type-mismatch" {
        description "Auth type mismatch.";
      }
    }
  }
}
enum "auth-failure" {
    description "Auth failure.";
}
enum "net-mask-mismatch" {
    description "Network mask mismatch.";
}
enum "hello-interval-mismatch" {
    description "Hello interval mismatch.";
}
enum "dead-interval-mismatch" {
    description "Dead interval mismatch.";
}
enum "option-mismatch" {
    description "Option mismatch.";
}
enum "mtu-mismatch" {
    description "MTU mismatch.";
}
enum "duplicate-router-id" {
    description "Duplicate Router ID.";
}
enum "no-error" {
    description "No error.";
}

description "Error code.";

description "This notification is sent when an 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 a 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 rt-types:timer-value-sectors16;
  description "Remaining time in current OSPF graceful restart interval when 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 inet:ip-address;
    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 AS-external limit (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 the
AS-external limit (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 modules specified in this document define a schema for data that 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 [RFC8446].

The NETCONF Access Control Model (NACM) [RFC8341] 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 ietf-ospf.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. Writable data node represent configuration of each instance, area, virtual link, sham-link, and interface. These correspond to the following schema nodes:

```
/ospf
/ospf/areas/

/ospf/areas/area[area-id]

/ospf/virtual-links/

/ospf/virtual-links/virtual-link[transit-area-id router-id]

/ospf/areas/area[area-id]/interfaces

/ospf/areas/area[area-id]/interfaces/interface[name]

/ospf/area/area[area-id]/sham-links

/ospf/area/area[area-id]/sham-links/sham-link[local-id remote-id]
```

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. For example, adding OSPF on any unprotected interface could allow an OSPF adjacency to be formed with an unauthorized and malicious neighbor. Once an adjacency is formed, traffic could be hijacked. As a simpler example, a Denial-of-Service attack could be mounted by changing the cost of an OSPF interface to be asymmetric such that a hard routing loop ensues. In general, unauthorized modification of most OSPF features will pose there own set of security risks and the "Security Considerations" in the respective reference RFCs should be consulted.

Some of the readable data nodes in the ietf-ospf.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. There is a separate Link State Database for each instance, area, virtual link, sham-link, and interface. These correspond to the following schema nodes:
Exposure of the Link State Database includes information beyond the scope of the OSPF router and this may be undesirable since exposure may facilitate other attacks. Additionally, in the case of an area LSDB, the complete IP network topology and, if deployed, the traffic engineering topology of the OSPF area can be reconstructed. Network operators may consider their topologies to be sensitive confidential data.

For OSPF authentication, configuration is supported via the specification of key-chains [RFC8177] or the direct specification of key and authentication algorithm. Hence, authentication 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.

Additionally, local specification of OSPF authentication keys and the associated authentication algorithm is supported for legacy implementations that do not support key-chains [RFC8177] It is RECOMMENDED that implementations migrate to key-chains due the seamless support of key and algorithm rollover, as well as, the hexadecimal key specification affording more key entropy, and encryption of keys using the Advanced Encryption Standard (AES) Key Wrap Padding Algorithm [RFC5649].

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 supports the "clear-neighbor" and "clear-database" RPCs. If access to either of these is compromised, they can result in temporary network outages be employed to mount DoS attacks.

The actual authentication key data (whether locally specified or part of a key-chain) is sensitive and needs to be kept secret from unauthorized parties; compromise of the key data would allow an

attacker to forge OSPF traffic that would be accepted as authentic, potentially compromising the entirety OSPF domain.

5. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-ospf
prefix: ospf
reference: RFC XXXX

6. Acknowledgements

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

Thanks to Tom Petch for last call review and improvement of the document organization.

Thanks to Alvaro Retana for AD comments.

Thanks to Benjamin Kaduk, Suresh Krishnan, and Roman Dannyliw for IESG review comments.

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

Author affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE’s concurrence with, or support for, the positions, opinions or viewpoints expressed. MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 18-3194.
7. References

7.1. Normative References

[I-D.ietf-bfd-yang]


Yeung, et al. Expires April 19, 2020


7.2. Informative References


Appendix A. Contributors’ Addresses

Dean Bogdanovic
Volta Networks, Inc.
EMail: dean@voltanet.io

Kiran Koushik Agrahara Sreenivasa
Verizon
500 W Dove Rd
Southlake, TX 76092
USA
EMail: kk@employees.org

Authors’ Addresses

Derek Yeung
Arrcus
EMail: derek@arrcus.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA
EMail: yingzhen.qu@futurewei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
EMail: zzhang@juniper.net

Ing-Wher Chen
The MITRE Corporation
EMail: ingwherchen@mitre.org
Abstract

OSPFv3 [RFC5340] defines an option field for router-LSAs known as a R-bit. If the R-bit is clear, an OSPFv3 router can participate in OSPF topology distribution without acting as a forwarder to forward the transit traffic. In such cases, an OSPF router would only accept traffic intended for local delivery. This draft defines R-bit functionality for OSPFv2 defined in [RFC2328].
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 2
2. Requirements Language ........................................ 2
3. H-bit Support .................................................. 3
4. SPF Modifications .............................................. 4
5. Auto Discovery and Backwards Compatibility ................. 5
6. IANA Considerations .......................................... 5
7. Security Considerations ...................................... 6
8. Acknowledgements ............................................. 6
9. Change Log ..................................................... 6
10. References .................................................. 6
   10.1. Normative References .................................... 6
   10.2. Informative References .................................. 6
Authors’ Addresses ............................................... 7

1. Introduction

OSPFv3 [RFC5340] defines an option field for router-LSAs known as a
R-bit. If the R-bit is clear, an OSPF router can participate in
OSPFv3 topology distribution without acting as a forwarder to forward
the transit traffic. In such cases, an OSPF router would only accept
traffic intended for local delivery.

This functionality is particularly useful for BGP Route Reflectors
known as virtual Route Reflectors (vRRs) that are not in the
forwarding path but are in central location such as data centers.
Such Route Reflectors typically are used for route distribution and
are not capable of forwarding data traffic. However, they need to
participate in the IGP routing for: 1) computing SPFs for Optimal
Route Reflection functionality defined in [I-D.ietf-idr-bgp-orr], and
2) resolving reachability for its Route Reflector Clients.

This draft defines R-bit functionality for OSPFv2 defined in
[RFC2328] by introducing a new Router LSA bit known as a "H-bit".

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to
be interpreted as described in [RFC2119] only when they appear in all
upper case. They may also appear in lower or mixed case as English
words, without any normative meaning.
3. H-bit Support

This draft defines a new Router-LSA bit known as a Host Bit or a H-bit. The H-bit indicates the OSPFv2’s capability of acting as a forwarder router. When set, the OSPFv2 router indicates that the forwarding capability is disabled. The bit value usage of the H-bit is reversed as opposed to the R-bit value defined in OSPFv3 [RFC5340] to support backward compatibility. The OSPFv2 Router LSA format is defined as:

```
|            LS age             |     Options   |       1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Link State ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    LS sequence number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         LS checksum                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   length                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|H|0|0|N|W|V|E|B|        0      |            # links            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Link ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Data                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     # TOS     |            metric             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TOS      |        0      |          TOS  metric          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Link ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Data                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

bit H
When set, an OSPFv2 router is a non-transit router and is incapable of acting as a forwarder.
When H-bit is set, an OSPFv2 router is a non-transit router and is incapable of acting as a forwarder. In this mode, the other OSPFv2 routers MUST not use the originating OSPFv2 router for the transit traffic, but they will use the OSPFv2 router for data traffic destined to that OSPFv2 router. An OSPFv2 router originating a Router LSA with the H-bit set SHOULD advertise its LINKS with MAX Link cost as defined in Section 3 of [RFC6987]. This is to increase the applicability of the H-bit in partial deployments where it is the responsibility of the operator to ensure that the H-bit does not result in routing loops.

When H-bit is set, only IPv4 prefixes associated with local interfaces MAY be advertised in summary LSAs. Non-local IPv4 prefixes, e.g., those advertised by other routers and installed during the SPF computation, MUST NOT be advertised in summary-LSAs. Likewise, when H-bit is set, only IPv4 prefixes associated with local interfaces MAY be advertised in AS-external LSAs. Non-local IPv4 prefixes, e.g., those exported from other routing protocols, MUST NOT be advertised in AS-external-LSAs. Finally, when H-bit is set, an ABR MUST advertise a consistent H-bit setting in its self-originated router-LSAs for all attached areas.

4. SPF Modifications

The SPF calculation described in section 16.1 [RFC2328] will be modified to assure that the routers originating router-LSAs with the H-bit set will not be used for transit traffic. Step 2 is modified as follows:

[Page 4]
2) Call the vertex just added to the tree vertex V. Examine the LSA associated with vertex V. This is a lookup in the Area A’s link state database based on the Vertex ID. If this is a router-LSA, and the H-bit of the router-LSA is set, and vertex V is not the root, then the router should not be used for transit and step (3) should be executed immediately. If this is a router-LSA, and bit V of the router-LSA (see Section A.4.2) is set, set Area A’s TransitCapability to TRUE. In any case, each link described by the LSA gives the cost to an adjacent vertex. For each described link, (say it joins vertex V to vertex W):

5. Auto Discovery and Backwards Compatibility

To avoid the possibility of any routing loops due to partial deployments, this draft defines a new OSPF Router Functional Capability known as a Host Support Capability. The value of this capability is a bit value to be assigned by IANA from OSPF Router Functional Capability Bits registry [I-D.ietf-ospf-rfc4970bis].

The Auto Discovery via announcement of the Host Support Functional Capability ensures that the H-bit functionality and its associated SPF changes SHOULD only take effect if all the routers in a given OSPF area support this functionality.

Implementations are encouraged to provide a knob to manually override enforcement of the H-bit functionality in partial deployment scenarios for cases where the topology guarantees that the router supporting the H-bit will not cause routing loops.

6. IANA Considerations

This draft defines a new Router LSA bit known as a H-bit. This draft requests IANA to 1) Create a new OSPF Router LSA bits registry and 2) assign a H-bit code type from the newly allocated OSPF Router LSA bit registry.

This draft defines a new Router Functional Capability known as a Host Support Functional Capability. This draft requests IANA to allocate
the value of this capability from the Router Functional Capability Bits TLV.

7. Security Considerations

This document introduces no new security considerations above and beyond those already specified in [RFC2328] and [RFC5340].

8. Acknowledgements

The authors would like to acknowledge Acee Lindem, Abhay Roy, David Ward, Burjiz Pithawala and Michael Barnes for their comments.

9. Change Log

Initial Version: April 23 2015

10. References

10.1. Normative References

[I-D.ietf-ospf-rfc4970bis]


10.2. Informative References

[I-D.ietf-idr-bgp-optimal-route-reflection]
Authors’ Addresses

Keyur Patel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124  95134
USA
Email: keyupate@cisco.com

Padma Pillay-Esnault
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124  95134
USA
Email: ppe@cisco.com

Manish Bhardwaj
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124  95134
USA
Email: manbhard@cisco.com

Serpil Bayraktar
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124  95134
USA
Email: serpil@cisco.com
Abstract

Some networks use tunnels for a variety of reasons. A large variety of tunnel types are defined and the ingress needs to select a type of tunnel which is supported by the egress. This document defines how to advertise egress tunnel capabilities in OSPF Router Information.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 1, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Some networks use tunnels for a variety of reasons, such as:

- Partial deployment of MPLS-SPRING as described in [I-D.xu-spring-islands-connection-over-ip], where IP tunnels are used between MPLS-SPRING-enabled routers so as to traverse non-MPLS routers.

- Partial deployment of MPLS-BIER as described in Section 6.9 of [I-D.ietf-bier-architecture], where IP tunnels are used between MPLS-BIER-capable routers so as to traverse non MPLS-BIER [I-D.ietf-bier-mpls-encapsulation] routers.
o Partial deployment of IPv6 (resp. IPv4) in IPv4 (resp. IPv6) networks as described in [RFC5565], where IPvx tunnels are used between IPvx-enabled routers so as to traverse non-IPvx routers.

o Remote Loop Free Alternate repair tunnels as described in [RFC7490], where tunnels are used between the Point of Local Repair and the selected PQ node.

The ingress needs to select a type of tunnel which is supported by the egress. This document describes how to use OSPF Router Information to advertise the egress tunnelling capabilities of nodes. In this document, OSPF means both OSPFv2 and OSPFv3.

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 [RFC4970].

3. Advertising Encapsulation Capability

Routers advertise their supported encapsulation type(s) by advertising a new TLV of the OSPF Router Information (RI) Opaque LSA [RFC4970], referred to as Encapsulation Capability TLV. This TLV is applicable to both OSPFv2 and OSPFv3. The Encapsulation Capability TLV SHOULD NOT appear more than once within a given OSPF Router Information (RI) Opaque LSA. The scope of the advertisement depends on the application but it is recommended that it SHOULD be domain-wide. The Type code of the Encapsulation Capability TLV is TBD1, the Length value is variable, and the Value field contains one or more Tunnel Encapsulation Type sub-TLVs. Each Encapsulation Type sub-TLV indicates a particular encapsulation format that the advertising router supports.

4. Tunnel Encapsulation Type

The Tunnel Encapsulation Type sub-TLV is structured as follows:
* Tunnel Type (2 octets): identifies the type of tunneling technology being signaled. This document defines the following types:

1. L2TPv3 over IP [RFC3931]: Type code=1;
2. GRE [RFC2784]: Type code=2;
3. Transmit tunnel endpoint [RFC5566]: Type code=3;
4. IPsec in Tunnel-mode [RFC5566]: Type code=4;
5. IP in IP tunnel with IPsec Transport Mode [RFC5566]: Type code=5;
6. MPLS-in-IP tunnel with IPsec Transport Mode [RFC5566]: Type code=6;
7. IP in IP [RFC2003] [RFC4213]: Type code=7;
8. VXLAN [I-D.ietf-bess-evpn-overlay]: Type code=8;
9. NVGRE [I-D.ietf-bess-evpn-overlay]: Type code=9;
10. MPLS [I-D.ietf-bess-evpn-overlay]: Type code=10;
11. MPLS-in-GRE [RFC4023]: Type code=11;
12. VxLAN GPE [RFC4023]: Type code=12;
13. MPLS-in-UDP [RFC7510]: Type code=13;
14. MPLS-in-UDP-with-DTLS [RFC7510]: Type code=14;
15. MPLS-in-L2TPv3 [RFC4017]: Type code=15;
16. GTP: Type code=16;
Unknown types are to be ignored and skipped upon receipt.

* Length (2 octets): unsigned integer indicating the total number of octets of the value field.

* Value (variable): zero or more Tunnel Encapsulation Attribute sub-TLVs as defined in Section 5.

5. Tunnel Encapsulation Attribute

The Tunnel Encapsulation Attribute sub-TLV is structured as as follows:

```
+-----------------------------------+
|      Sub-TLV Type (1 Octet)       |
+-----------------------------------+
|     Sub-TLV Length (1 Octet)      |
+-----------------------------------+
|     Sub-TLV Value (Variable)      |
+-----------------------------------+
```

* Sub-TLV Type (1 octet): each sub-TLV type defines a certain property about the tunnel TLV that contains this sub-TLV. The following are the types defined in this document:

1. Encapsulation Parameters: sub-TLV type = 1; (See Section 5.1)
2. Encapsulated Protocol: sub-TLV type = 2; (See Section 5.2)
3. End Point: sub-TLV type = 3; (See Section 5.3)
4. Color: sub-TLV type = 4; (See Section 5.4)

* Sub-TLV Length (1 octet): unsigned integer indicating the total number of octets of the sub-TLV value field.

* Sub-TLV Value (variable): encodings of the value field depend on the sub-TLV type as enumerated above. The following sub-sections define the encoding in detail.

Any unknown sub-TLVs MUST be ignored and skipped. However, if the TLV is understood, the entire TLV MUST NOT be ignored just because it contains an unknown sub-TLV.
If a sub-TLV is erroneous, this specific Tunnel Encapsulation MUST be ignored and skipped. However, others Tunnel Encapsulations MUST be considered.

5.1. Tunnel Parameters sub-TLV

This sub-TLV has its format defined in [RFC5512] under the name Encapsulation sub-TLV.

5.2. Encapsulated Protocol sub-TLV

This sub-TLV has its format defined in [RFC5512] under the name Protocol Type.

5.3. End Point sub-TLV

The value field carries the Network Address to be used as tunnel destination address.

If length is 4, the Address Family (AFI) is IPv4.

If length is 16, the Address Family (AFI) is IPv6.

5.4. Color sub-TLV

The valued field is a 4 octets opaque unsigned integer.

The color value is user defined and configured locally on the routers. It may be used by the service providers to define policies.

6. IANA Considerations

6.1. OSPF Router Information

This document requests IANA to allocate a new code point from registry OSPF Router Information (RI).

<table>
<thead>
<tr>
<th>Value</th>
<th>TLV Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>Tunnel Capabilities</td>
<td>This document</td>
</tr>
</tbody>
</table>

6.2. IGP Tunnel Encapsulation Types Registry

This document requests IANA to create a new registry "IGP Tunnel Encapsulation Types" with the following registration procedure:
Registry Name: IGP Tunnel Encapsulation Type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>This document</td>
</tr>
<tr>
<td>1</td>
<td>L2TPv3 over IP</td>
<td>This document</td>
</tr>
<tr>
<td>2</td>
<td>GRE</td>
<td>This document</td>
</tr>
<tr>
<td>3</td>
<td>Transmit tunnel endpoint</td>
<td>This document</td>
</tr>
<tr>
<td>4</td>
<td>IPsec in Tunnel-mode</td>
<td>This document</td>
</tr>
<tr>
<td>5</td>
<td>IP in IP tunnel with IPsec Transport Mode</td>
<td>This document</td>
</tr>
<tr>
<td>6</td>
<td>MPLS-in-IP tunnel with IPsec Transport Mode</td>
<td>This document</td>
</tr>
<tr>
<td>7</td>
<td>IP in IP</td>
<td>This document</td>
</tr>
<tr>
<td>8</td>
<td>VXLAN</td>
<td>This document</td>
</tr>
<tr>
<td>9</td>
<td>NVGRE</td>
<td>This document</td>
</tr>
<tr>
<td>10</td>
<td>MPLS</td>
<td>This document</td>
</tr>
<tr>
<td>11</td>
<td>MPLS-in-GRE</td>
<td>This document</td>
</tr>
<tr>
<td>13</td>
<td>MPLS-in-UDP</td>
<td>This document</td>
</tr>
<tr>
<td>14</td>
<td>MPLS-in-UDP-with-DTLS</td>
<td>This document</td>
</tr>
<tr>
<td>15</td>
<td>MPLS-in-L2TPv3</td>
<td>This document</td>
</tr>
<tr>
<td>16</td>
<td>GTP</td>
<td>This document</td>
</tr>
<tr>
<td>17-250</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>251-254</td>
<td>Experimental</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>Reserved</td>
<td>This document</td>
</tr>
</tbody>
</table>

Assignments of Encapsulation Types are via Standards Action [RFC5226].

6.3. IGP Tunnel Encapsulation Attribute Types Registry

This document requests IANA to create a new registry "IGP Tunnel Encapsulation Attribute Types" with the following registration procedure:

Registry Name: IGP Tunnel Encapsulation Attribute Types.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>This document</td>
</tr>
<tr>
<td>1</td>
<td>Encapsulation parameters</td>
<td>This document</td>
</tr>
<tr>
<td>2</td>
<td>Protocol</td>
<td>This document</td>
</tr>
<tr>
<td>3</td>
<td>End Point</td>
<td>This document</td>
</tr>
<tr>
<td>4</td>
<td>Color</td>
<td>This document</td>
</tr>
<tr>
<td>5-250</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>251-254</td>
<td>Experimental</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>Reserved</td>
<td>This document</td>
</tr>
</tbody>
</table>
Assignments of Encapsulation Types are via Standards Action [RFC5226].

7. Security Considerations

Security considerations applicable to softwires can be found in the mesh framework [RFC5565]. In general, security issues of the tunnel protocols signaled through this IGP capability extension are inherited.

If a third party is able to modify any of the information that is used to form encapsulation headers, to choose a tunnel type, or to choose a particular tunnel for a particular payload type, user data packets may end up getting misrouted, misdelivered, and/or dropped.

Security considerations for the base OSPF protocol are covered in [RFC2328] and [RFC5340].

8. Acknowledgements

This document is partially inspired by [RFC5512].

The authors would like to thank Carlos Pignataro and Karsten Thomann for their valuable comments on this draft.

9. References

9.1. Normative References


9.2. Informative References

[I-D.ietf-bess-evpn-overlay]

[I-D.ietf-bier-architecture]

[I-D.ietf-bier-mpls-encapsulation]

[I-D.xu-spring-islands-connection-over-ip]

[IANA-OSPFv2]


Authors’ Addresses

Xiaohu Xu (editor)
Huawei
Email: xuxiaohu@huawei.com

Bruno Decraene (editor)
Orange
Email: bruno.decreaene@orange.com

Robert Raszuk
Mirantis Inc.
Email: robert@raszuk.net

Uma Chunduri
Ericsson
Email: uma.chunduri@ericsson.com