Network Working Group
Internet-Draft
Updates: 5340, 5838 (if approved)
Intended status: Standards Track
Expires: July 29, 2018

A. Lindem
A. Roy
Cisco Systems
D. Goethals
Nokia
V. Reddy Vallem

F. Baker
January 25, 2018

OSPFv3 LSA Extendibility
draft-ietf-ospf-ospfv3-lsa-extend-23.txt

Abstract

OSPFv3 requires functional extension beyond what can readily be done with the fixed-format Link State Advertisement (LSA) as described in RFC 5340. Without LSA extension, attributes associated with OSPFv3 links and advertised IPv6 prefixes must be advertised in separate LSAs and correlated to the fixed-format LSAs. This document extends the LSA format by encoding the existing OSPFv3 LSA information in Type-Length-Value (TLV) tuples and allowing advertisement of additional information with additional TLVs. Backward compatibility mechanisms are also described.

This document updates RFC 5340, "OSPF for IPv6", and RFC 5838, "Support of Address Families in OSPFv3" by providing TLV-based encodings for the base OSPFv3 unicast support and OSPFv3 address family support.

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 July 29, 2018.

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

Table of Contents

1. Introduction 3
  1.1. Requirements notation 4
  1.2. OSPFv3 LSA Terminology 4
2. OSPFv3 Extended LSA Types 4
3. OSPFv3 Extended LSA TLVs 5
  3.1. Prefix Options Extensions 6
    3.1.1. N-bit Prefix Option 6
  3.2. Router-Link TLV 7
  3.3. Attached-Routers TLV 8
  3.4. Inter-Area-Prefix TLV 10
  3.5. Inter-Area-Router TLV 11
  3.6. External-Prefix TLV 12
  3.7. Intra-Area-Prefix TLV 13
  3.8. IPv6 Link-Local Address TLV 14
  3.9. IPv4 Link-Local Address TLV 15
  3.10. IPv6-Forwarding-Address Sub-TLV 16
  3.11. IPv4-Forwarding-Address Sub-TLV 16
  3.12. Route-Tag Sub-TLV 17
4. OSPFv3 Extended LSAs 17
  4.1. OSPFv3 E-Router-LSA 17
  4.2. OSPFv3 E-Network-LSA 19
  4.3. OSPFv3 E-Inter-Area-Prefix-LSA 20
  4.4. OSPFv3 E-Inter-Area-Router-LSA 21
  4.5. OSPFv3 E-AS-External-LSA 22
  4.6. OSPFv3 E-NSSA-LSA 23
  4.7. OSPFv3 E-Link-LSA 24
  4.8. OSPFv3 E-Intra-Area-Prefix-LSA 26
5. Malformed OSPFv3 Extended LSA Handling 27
6. LSA Extension Backward Compatibility 27
  6.1. Full Extended LSA Migration 27
  6.2. Extended LSA Sparse-Mode Backward Compatibility 28

6.3. LSA TLV Processing Backward Compatibility 28
7. Security Considerations 29
8. IANA Considerations 29
8.1. OSPFv3 Extended-LSA TLV Registry 29
8.2. OSPFv3 Extended-LSA sub-TLV Registry 30
9. Contributors 31
10. References 31
10.1. Normative References 31
10.2. Informative References 31
Appendix A. Appendix A - Global Configuration Parameters 32
Appendix B. Appendix B - Area Configuration Parameters 32
Appendix C. Acknowledgments 33
Authors' Addresses 33

1. Introduction

OSPFv3 requires functional extension beyond what can readily be done with the fixed-format Link State Advertisement (LSA) as described in RFC 5340 [OSPFV3]. Without LSA extension, attributes associated with OSPFv3 links and advertised IPv6 prefixes must be advertised in separate LSAs and correlated to the fixed-format LSAs. This document extends the LSA format by encoding the existing OSPFv3 LSA information in Type-Length-Value (TLV) tuples and allowing advertisement of additional information with additional TLVs. Backward compatibility mechanisms are also described.

This document updates RFC 5340, "OSPF for IPv6", and RFC 5838, "Support of Address Families in OSPFv3" by providing TLV-based encodings for the base OSPFv3 support [OSPFV3] and OSPFv3 address family support [OSPFV3-AF].

A similar extension was previously proposed in support of multi-topology routing. Additional requirements for OSPFv3 LSA extension include source/destination routing, route tagging, and others.

A final requirement is to limit the changes to OSPFv3 to those necessary for TLV-based LSAs. For the most part, the semantics of existing OSPFv3 LSAs are retained for their TLV-based successor LSAs described herein. Additionally, encoding details, e.g., the representation of IPv6 prefixes as described in section A.4.1 in RFC 5340 [OSPFV3], have been retained. This requirement was included to increase the expedience of IETF adoption and deployment.

The following aspects of OSPFv3 LSA extension are described:

1. Extended LSA Types

2. Extended LSA TLVs

3. Extended LSA Formats

4. Backward Compatibility

1.1. Requirements notation

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. OSPFv3 LSA Terminology

The TLV-based OSPFv3 LSAs described in this document will be referred to as Extended LSAs. The OSPFv3 fixed-format LSAs [OSPFV3] will be referred to as Legacy LSAs.

2. OSPFv3 Extended LSA Types

In order to provide backward compatibility, new LSA codes must be allocated. There are eight fixed-format LSAs defined in RFC 5340 [OSPFV3]. For ease of implementation and debugging, the LSA function codes are the same as the fixed-format LSAs only with 32, i.e., 0x20, added. The alternative to this mapping was to allocate a bit in the LS Type indicating the new LSA format. However, this would have used one half the LSA function code space for the migration of the eight original fixed-format LSAs. For backward compatibility, the U-bit MUST be set in LS Type so that the LSAs will be flooded by OSPFv3 routers that do not understand them.

LSA function code	LS Type	Description
33	0xA021	E-Router-LSA
34	0xA022	E-Network-LSA
35	0xA023	E-Inter-Area-Prefix-LSA
36	0xA024	E-Inter-Area-Router-LSA
37	0xC025	E-AS-External-LSA
38	N/A	Unused (Not to be allocated)
39	0xA027	E-Type-7-LSA
40	0x8028	E-Link-LSA
41	0xA029	E-Intra-Area-Prefix-LSA

OSPFv3 Extended LSA Types

3. OSPFv3 Extended LSA TLVs

The format of the TLVs within the body of the extended LSAs is the same as the format used by the Traffic Engineering Extensions to OSPF [TE]. The variable TLV section consists of one or more nested Type/Length/Value (TLV) tuples. Nested TLVs are also referred to as sub-TLVs. The format of each TLV is:

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value...                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Format

The Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-byte value would have the length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV.

This document defines the following top-level TLV types:

- o 0 - Reserved
- o 1 - Router-Link TLV
- o 2 - Attached-Routers TLV
- o 3 - Inter-Area Prefix TLV
- o 4 - Inter-Area Router TLV
- o 5 - External Prefix TLV
- o 6 - Intra-Area Prefix TLV
- o 7 - IPv6 Link-Local Address TLV
- o 8 - IPv4 Link-Local Address TLV

Additionally, this document defines the following sub-TLV types:

- o 0 - Reserved
- o 1 - IPv6 Forwarding Address sub-TLV
- o 2 - IPv4 Forwarding Address sub-TLV
- o 3 - Route Tag sub-TLV

In general, TLVs and sub-TLVs MAY occur in any order and the specification should define whether the TLV or sub-TLV is required and the behavior when there are multiple occurrences of the TLV or sub-TLV. While this document only describes the usage of TLVs and Sub-TLVs, Sub-TLVs may be nested to any level as long as the Sub-TLVs are fully specified in the specification for the subsuming Sub-TLV.

For backward compatibility, an LSA is not considered malformed from a TLV perspective unless either a required TLV is missing or a specified TLV is less than the minimum required length. Refer to Section 6.3 for more information on TLV backward compatibility.

3.1. Prefix Options Extensions

The prefix options are extended from Appendix A.4.1.1 [OSPFV3]. The applicability of the LA-bit is expanded and it SHOULD be set in Inter-Area-Prefix-TLVs and MAY be set in External-Prefix-TLVs when the advertised host IPv6 address, i.e., PrefixLength = 128, is an interface address. In RFC 5340, the LA-bit is only set in Intra-Area-Prefix-LSAs (Section 4.4.3.9 in [OSPFV3]). This will allow a stable address to be advertised without having to configure a separate loopback address in every OSPFv3 area.

3.1.1. N-bit Prefix Option

Additionally, the N-bit prefix option is defined. The figure below shows the position of the N-bit in the prefix options (pending IANA allocation). This corresponds to the value 0x20.

```
  0  1  2  3  4  5  6  7
+--+--+--+--+--+--+--+--+
|  |  | N|DN| P| x|LA|NU|
+--+--+--+--+--+--+--+--+
```

The Prefix Options field

The N-bit is set in PrefixOptions for a host address (PrefixLength=128) that identifies the advertising router. While it is similar to the LA-bit, there are two differences. The advertising router MAY choose NOT to set the N-bit even when the above conditions are met. If the N-bit is set and the PrefixLength is NOT 128, the N-bit MUST be ignored. Additionally, the N-bit is propagated in the PrefixOptions when an OSPFv3 Area Border Router (ABR) originates an Inter-Area-Prefix-LSA for an Intra-Area route which has the N-bit set in the PrefixOptions. Similarly, the N-bit is propagated in the PrefixOptions when an OSPFv3 NSSA ABR originates an E-AS-External-LSA corresponding to an NSSA route as described in section 3 of RFC 3101 ([NSSA]). The N-bit is added to the Inter-Area-Prefix-TLV (Section 3.4), External-Prefix-TLV (Section 3.6), and Intra-Area-Prefix-TLV (Section 3.7). The N-bit is used as hint to identify the preferred address to reach the advertising OSPFv3 router. This would be in contrast to an Anycast Address [IPV6-ADDRESS-ARCH] which could also be a local address with the LA-bit set. It is useful for applications such as identifying the prefixes corresponding to Node Segment Identifiers (SIDs) in Segment Routing [SEGMENT-ROUTING]. There may be future applications requiring selection of a prefix associated with an OSPFv3 router.

3.2. Router-Link TLV

The Router-Link TLV defines a single router link and the field definitions correspond directly to links in the OSPFv3 Router-LSA, section A.4.3, [OSPFV3]. The Router-Link TLV is only applicable to the E-Router-LSA (Section 4.1). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          1 (Router-Link)      |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |       0       |           Metric              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Interface ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Neighbor Interface ID                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Neighbor Router ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             sub-TLVs                          .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Router-Link TLV

3.3. Attached-Routers TLV

The Attached-Routers TLV defines all the routers attached to an OSPFv3 multi-access network. The field definitions correspond directly to content of the OSPFv3 Network-LSA, section A.4.4, [OSPFV3]. The Attached-Routers TLV is only applicable to the E-Network-LSA (Section 4.2). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        2 (Attached-Routers)   |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Adjacent Neighbor Router ID                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.              Additional Adjacent Neighbors                    .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Attached-Routers TLV

There are two reasons for not having a separate TLV or sub-TLV for each adjacent neighbor. The first is to discourage using the E-Network-LSA for more than its current role of solely advertising the routers attached to a multi-access network. The router’s metric as well as the attributes of individual attached routers should be

advertised in their respective E-Router-LSAs. The second reason is that there is only a single E-Network-LSA per multi-access link with the Link State ID set to the Designated Router’s Interface ID and, consequently, compact encoding has been chosen to decrease the likelihood that the size of the E-Network-LSA will require IPv6 fragmentation when advertised in an OSPFv3 Link State Update packet.

3.4. Inter-Area-Prefix TLV

The Inter-Area-Prefix TLV defines a single OSPFV3 inter-area prefix. The field definitions correspond directly to the content of an OSPFv3 IPv6 Prefix as defined in Section A.4.1, [OSPFV3] and an OSPFv3 Inter-Area-Prefix-LSA, as defined in section A.4.5, [OSPFV3]. Additionally, the PrefixOptions are extended as described in Section 3.1. The Inter-Area-Prefix TLV is only applicable to the E-Inter-Area-Prefix-LSA (Section 4.3). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       3 (Inter-Area Prefix)   |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0        |                  Metric                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PrefixLength  | PrefixOptions |              0                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Address Prefix                         |
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            sub-TLVs                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Inter-Area Prefix TLV

3.5. Inter-Area-Router TLV

The Inter-Area-Router TLV defines a single OSPFv3 Autonomous System Boundary Router (ASBR) reachable in another area. The field definitions correspond directly to the content of an OSPFv3 Inter-Area-Router-LSA, as defined in section A.4.6, [OSPFV3]. The Inter-Area-Router TLV is only applicable to the E-Inter-Area-Router-LSA (Section 4.4). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       4 (Inter-Area Router)   |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0        |                Options                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0        |                Metric                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Destination Router ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            sub-TLVs                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Inter-Area Router TLV

3.6. External-Prefix TLV

The External-Prefix TLV defines a single OSPFv3 external prefix. With the exception of omitted fields noted below, the field definitions correspond directly to the content of an OSPFv3 IPv6 Prefix as defined in Section A.4.1, [OSPFV3] and an OSPFv3 AS-External-LSA, as defined in section A.4.7, [OSPFV3]. The External-Prefix TLV is only applicable to the E-AS-External-LSA (Section 4.5) and the E-NSSA-LSA (Section 4.6). Additionally, the PrefixOptions are extended as described in Section 3.1. Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       5 (External Prefix)     |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         |E| | |                Metric                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PrefixLength  | PrefixOptions |              0                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Address Prefix                         |
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            sub-TLVs                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

External Prefix TLV

In the External-Prefix TLV, the optional IPv6/IPv4 Forwarding Address and External Route Tag are now sub-TLVs. Given the Referenced LS type and Referenced Link State ID from the AS-External-LSA have never been used or even specified, they have been omitted from the External Prefix TLV. If there were ever a requirement for a referenced LSA, it could be satisfied with a sub-TLV.

The following sub-TLVs are defined for optional inclusion in the External Prefix TLV:

- o 1 - IPv6 Forwarding Address sub-TLV (Section 3.10)

- o 2 - IPv4 Forwarding Address sub-TLV (Section 3.11)

- o 3 - Route Tag sub-TLV (Section 3.12)

3.7. Intra-Area-Prefix TLV

The Intra-Area-Prefix TLV defines a single OSPFv3 intra-area prefix. The field definitions correspond directly to the content of an OSPFv3 IPv6 Prefix as defined in Section A.4.1, [OSPFV3] and an OSPFv3 Link-LSA, as defined in section A.4.9, [OSPFV3]. The Intra-Area-Prefix TLV is only applicable to the E-Link-LSA (Section 4.7) and the E-Intra-Area-Prefix-LSA (Section 4.8). Additionally, the PrefixOptions are extended as described in Section 3.1. Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       6 (Intra-Area Prefix)   |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0        |                  Metric                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PrefixLength  | PrefixOptions |              0                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Address Prefix                         |
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            sub-TLVs                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Intra-Area Prefix TLV

3.8. IPv6 Link-Local Address TLV

The IPv6 Link-Local Address TLV is to be used with IPv6 address families as defined in [OSPFV3-AF]. The IPv6 Link-Local Address TLV is only applicable to the E-Link-LSA (Section 4.7). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  7 (IPv6 Local-Local Address) |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-            IPv6 Link-Local Interface Address                -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            sub-TLVs                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 Link-Local Address TLV

3.9. IPv4 Link-Local Address TLV

The IPv4 Link-Local Address TLV is to be used with IPv4 address families as defined in [OSPFV3-AF]. The IPv4 Link-Local Address TLV is only applicable to the E-Link-LSA (Section 4.7). Inclusion in other Extended LSAs MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  8 (IPv4 Local-Local Address) |       TLV Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IPv4 Link-Local Interface Address                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             sub-TLVs                          .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Link-Local Address TLV

3.10. IPv6-Forwarding-Address Sub-TLV

The IPv6 Forwarding Address TLV has identical semantics to the optional forwarding address in section A.4.7 of [OSPFV3]. The IPv6 Forwarding Address TLV is applicable to the External-Prefix TLV (Section 3.6). Specification as a sub-TLV of other TLVs is not defined herein. The sub-TLV is optional and the first specified instance is used as the Forwarding Address as defined in [OSPFV3]. Instances subsequent to the first MUST be ignored.

The IPv6 Forwarding Address TLV is to be used with IPv6 address families as defined in [OSPFV3-AF] It MUST be ignored for other address families. The IPv6 Forwarding Address TLV length must meet minimum length (16 octets) or it will be considered malformed as described in Section 6.3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       1 - Forwarding Address  |          sub-TLV Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-                Forwarding Address                           -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 Forwarding Address TLV

3.11. IPv4-Forwarding-Address Sub-TLV

The IPv4 Forwarding Address TLV has identical semantics to the optional forwarding address in section A.4.7 of [OSPFV3]. The IPv4 Forwarding Address TLV is The IPv4 Forwarding Address TLV is applicable to the External-Prefix TLV (Section 3.6). Specification as a sub-TLV of other TLVs is not defined herein. The sub-TLV is optional and the first specified instance is used as the Forwarding Address as defined in [OSPFV3]. Instances subsequent to the first MUST be ignored.

The IPv4 Forwarding Address TLV is to be used with IPv4 address families as defined in [OSPFV3-AF] It MUST be ignored for other address families. The IPv4 Forwarding Address TLV length must meet minimum length (4 octets) or it will be considered malformed as described in Section 6.3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       2 - Forwarding Address  |          sub-TLV Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Forwarding Address                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Forwarding Address TLV

3.12. Route-Tag Sub-TLV

The optional Route Tag sub-TLV has identical semantics to the optional External Route Tag in section A.4.7 of [OSPFV3]. The Route Tag sub-TLV is applicable to the External-Prefix TLV (Section 3.6). Specification as a sub-TLV of other TLVs is not defined herein. The sub-TLV is optional and the first specified instance is used as the Route Tag as defined in [OSPFV3]. Instances subsequent to the first MUST be ignored.

The Route Tag TLV length must meet minimum length (4 octets) or it will be considered malformed as described in Section 6.3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       3 - Route Tag           |          sub-TLV Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Route Tag                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Route Tag Sub-TLV

4. OSPFv3 Extended LSAs

This section specifies the OSPFv3 Extended LSA formats and encoding. The Extended OSPFv3 LSAs corresponded directly to the original OSPFv3 LSAs specified in [OSPFV3].

4.1. OSPFv3 E-Router-LSA

The E-Router-LSA has an LS Type of 0xA021 and has the same base information content as the Router-LSA defined in section A.4.3 of [OSPFV3]. However, unlike the existing Router-LSA, it is fully extendable and represented as TLVs.

```
 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
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LS Age              |1|0|1|         0x21            |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Link State ID                           |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS Sequence Number                        |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        LS Checksum            |            Length             |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  0  |Nt|x|V|E|B|            Options                           |
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             TLVs                              .
.                                                               .
+-+-+-+--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Extended Router-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Router-LSA. Initially, only the top-level Router-Link TLV Section 3.2 is applicable and an E-Router-LSA may include multiple Router-Link TLVs. Like the existing Router-LSA, the LSA length is used to determine the end of the LSA including TLVs. Depending on the implementation, it is perfectly valid for an E-Router-LSA to not contain any Router-Link TLVs. However, this would imply that the OSPFv3 router doesn’t have any adjacencies in the corresponding area and is forming an adjacency or adjacencies over unnumbered link(s). Note that no E-Router-LSA stub link is advertised for an unnumbered link.

4.2. OSPFv3 E-Network-LSA

The E-Network-LSA has an LS Type of 0xA022 and has the same base information content as the Network-LSA defined in section A.4.4 of [OSPFV3]. However, unlike the existing Network-LSA, it is fully extendable and represented as TLVs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          LS Age               |1|0|1|         0x22            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Advertising Router                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    LS Sequence Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LS Checksum             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |            Options                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            TLVs                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-Network-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Network-LSA. Like the existing Network-LSA, the LSA length is used to determine the end of the LSA including TLVs. Initially, only the top-level Attached-Routers TLV Section 3.3 is applicable. If the Attached-Router TLV is not included in the E-Network-LSA, it is treated as malformed as described in Section 5. Instances of the Attached-Router TLV subsequent to the first MUST be ignored.

4.3. OSPFv3 E-Inter-Area-Prefix-LSA

The E-Inter-Area-Prefix-LSA has an LS Type of 0xA023 and has the same base information content as the Inter-Area-Prefix-LSA defined in section A.4.5 of [OSPFV3]. However, unlike the existing Inter-Area-Prefix-LSA, it is fully extendable and represented as TLVs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          LS Age               |1|0|1|         0x23            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Advertising Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LS Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LS Checksum             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             TLVs                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-Inter-Area-Prefix-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Inter-Area-Prefix-LSA. In order to retain compatibility and semantics with the current OSPFv3 specification, each Inter-Area-Prefix LSA MUST contain a single Inter-Area Prefix TLV. This will facilitate migration and avoid changes to functions such as incremental SPF computation.

Like the existing Inter-Area-Prefix-LSA, the LSA length is used to determine the end of the LSA including TLV. Initially, only the top-level Inter-Area-Prefix TLV (Section 3.4) is applicable. If the Inter-Area-Prefix TLV is not included in the E-Inter-Area-Prefix-LSA, it is treated as malformed as described in Section 5. Instances of the Inter-Area-Prefix TLV subsequent to the first MUST be ignored.

4.4. OSPFv3 E-Inter-Area-Router-LSA

The E-Inter-Area-Router-LSA has an LS Type of 0xA024 and has the same base information content as the Inter-Area-Router-LSA defined in section A.4.6 of [OSPFV3]. However, unlike the Inter-Area-Router-LSA, it is fully extendable and represented as TLVs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          LS Age               |1|0|1|         0x24            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Advertising Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LS Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LS Checksum             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             TLVs                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-Inter-Area-Router-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Inter-Area-Router-LSA. In order to retain compatibility and semantics with the current OSPFv3 specification, each Inter-Area-Router LSA MUST contain a single Inter-Area Router TLV. This will facilitate migration and avoid changes to functions such as incremental SPF computation.

Like the existing Inter-Area-Router-LSA, the LSA length is used to determine the end of the LSA including TLV. Initially, only the top-level Inter-Area-Router TLV (Section 3.5) is applicable. If the Inter-Area-Router TLV is not included in the E-Inter-Area-Router-LSA, it is treated as malformed as described in Section 5. Instances of the Inter-Area-Router TLV subsequent to the first MUST be ignored.

4.5. OSPFv3 E-AS-External-LSA

The E-AS-External-LSA has an LS Type of 0xC025 and has the same base information content as the AS-External-LSA defined in section A.4.7 of [OSPFV3]. However, unlike the existing AS-External-LSA, it is fully extendable and represented as TLVs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          LS Age               |1|1|0|         0x25            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Advertising Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LS Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LS Checksum             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             TLVs                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-AS-External-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the AS-External-LSA. In order to retain compatibility and semantics with the current OSPFv3 specification, each LSA MUST contain a single External Prefix TLV. This will facilitate migration and avoid changes to OSPFv3 processes such as incremental SPF computation.

Like the existing AS-External-LSA, the LSA length is used to determine the end of the LSA including sub-TLVs. Initially, only the top-level External-Prefix TLV (Section 3.6) is applicable. If the External-Prefix TLV is not included in the E-External-AS-LSA, it is treated as malformed as described in Section 5. Instances of the External-Prefix TLV subsequent to the first MUST be ignored.

4.6. OSPFv3 E-NSSA-LSA

The E-NSSA-LSA will have the same format and TLVs as the Extended AS-External-LSA Section 4.5. This is the same relationship as exists between the NSSA-LSA defined in section A.4.8 of [OSPFV3], and the AS-External-LSA. The NSSA-LSA will have type 0xA027 which implies area flooding scope. Future requirements may dictate that supported TLVs differ between the E-AS-External-LSA and the E-NSSA-LSA. However, future requirements are beyond the scope of this document.

4.7. OSPFv3 E-Link-LSA

The E-Link-LSA has an LS Type of 0x8028 and will have the same base information content as the Link-LSA defined in section A.4.9 of [OSPFV3]. However, unlike the existing Link-LSA, it is extendable and represented as TLVs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          LS Age               |1|0|0|         0x28            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Advertising Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LS Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LS Checksum             |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Rtr Priority  |                Options                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            TLVs                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-Link-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Link-LSA.

Only the Intra-Area-Prefix TLV (Section 3.7), IPv6 Link-Local Address TLV (Section 3.8), and IPv4 Link-Local Address TLV (Section 3.9) are applicable to the E-Link-LSA. Like the Link-LSA, the E-Link-LSA affords advertisement of multiple intra-area prefixes. Hence, multiple Intra-Area Prefix TLVs (Section 3.7) may be specified and the LSA length defines the end of the LSA including all TLVs.

A single instance of the IPv6 Link-Local Address TLV (Section 3.8) SHOULD be included in the E-Link-LSA. Instances following the first MUST be ignored. For IPv4 address families as defined in [OSPFV3-AF], this TLV MUST be ignored.

Similarly, only a single instance of the IPv4 Link-Local Address TLV (Section 3.9) SHOULD be included in the E-Link-LSA. Instances

following the first MUST be ignored. For OSPFv3 IPv6 address families as defined in [OSPFV3-AF], this TLV SHOULD be ignored.

If the IPv4/IPv6 Link-Local Address TLV corresponding to the OSPFv3 Address Family is not included in the E-Link-LSA, it is treated as malformed as described in Section 5.

Future specifications may support advertisement of routing and topology information for multiple address families. However, this is beyond the scope of this document.

4.8. OSPFv3 E-Intra-Area-Prefix-LSA

The E-Intra-Area-Prefix-LSA has an LS Type of 0xA029 and has the same base information content as the Intra-Area-Prefix-LSA defined in section A.4.10 of [OSPFV3] except for the Referenced LS Type. However, unlike the Intra-Area-Prefix-LSA, it is fully extendable and represented as TLVs. The Referenced LS Type MUST be either an E-Router-LSA (0xA021) or an E-Network-LSA (0xA022).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LS Age              |1|0|1|         0x29            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link State ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Advertising Router                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LS Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        LS Checksum            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0                       |     Referenced LS Type        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Referenced Link State ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Referenced Advertising Router                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                            TLVs                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E-Intra-Area-Prefix-LSA

Other than having a different LS Type, all LSA Header fields are the same as defined for the Intra-Area-Prefix-LSA.

Like the Intra-Area-Prefix-LSA, the E-Intra-Area-Link-LSA affords advertisement of multiple intra-area prefixes. Hence, multiple Intra-Area Prefix TLVs may be specified and the LSA length defines the end of the LSA including all TLVs.

5. Malformed OSPFv3 Extended LSA Handling

Extended LSAs that have inconsistent length or other encoding errors, as described herein, MUST NOT be installed in the Link State Database, acknowledged, or flooded. Reception of malformed LSAs SHOULD be counted and/or logged for examination by the administrator of the OSPFv3 Routing Domain. Note that for the purposes of length validation, a TLV or Sub-TLV should not be considered invalid unless the length exceeds the length of the LSA or does not meet the minimum length requirements. This allows for Sub-TLVs to be added as described in Section 6.3.

Additionally, an LSA MUST be considered malformed if it does not include all of the required TLVs and Sub-TLVs.

6. LSA Extension Backward Compatibility

In the context of this document, backward compatibility is solely related to the capability of an OSPFv3 router to receive, process, and originate the TLV-based LSAs defined herein. Unrecognized TLVs and sub-TLVs are ignored. Backward compatibility for future OSPFv3 extensions utilizing the TLV-based LSAs is out of scope and must be covered in the documents describing those extensions. Both full and, if applicable, partial deployment SHOULD be specified for future TLV-based OSPFv3 LSA extensions.

6.1. Full Extended LSA Migration

If ExtendedLSASupport is enabled Appendix A, OSPFv3 Extended LSAs will be originated and used for the SPF computation. Individual OSPF Areas can be migrated separately with the Legacy AS-External LSAs being originated and used for the SPF computation. This is accomplished by enabled AreaExtendedLSASupport Appendix B.

An OSPFv3 routing domain or area may be non-disruptively migrated using separate OSPFv3 instances for the extended LSAs. Initially, the OSPFv3 instances with ExtendedLSASupport will have a lower preference, i.e., higher administrative distance, than the OSPFv3 instances originating and using the Legacy LSAs. Once the routing domain or area is fully migrated and the OSPFv3 Routing Information Bases (RIB) have been verified, the OSPFv3 instances using the extended LSAs can be given preference. When this has been completed and the routing within the OSPF routing domain or area has been verified, the original OSPFv3 instance using Legacy LSAs can be removed.

6.2. Extended LSA Sparse-Mode Backward Compatibility

In this mode, OSPFv3 will use the Legacy LSAs for the SPF computation and will only originate extended LSAs when LSA origination is required in support of additional functionality. Furthermore, those extended LSAs will only include the top-level TLVs (e.g., Router-Link TLVs or Inter-Area TLVs) which require further specification for that new functionality. However, if a top-level TLV is advertised, it MUST include required Sub-TLVs or it will be considered malformed as described in Section 5. Hence, this mode of compatibility is known as "sparse-mode". The advantage of sparse-mode is that functionality utilizing the OSPFv3 extended LSAs can be added to an existing OSFPv3 routing domain without the requirement for migration. In essence, this compatibility mode is very much like the approach taken for OSPFv2 [OSPF-PREFIX-LINK]. As with all the compatibility modes, backward compatibility for the functions utilizing the extended LSAs must be described in the IETF documents describing those functions.

6.3. LSA TLV Processing Backward Compatibility

This section defines the general rules for processing LSA TLVs. To ensure compatibility of future TLV-based LSA extensions, all implementations MUST adhere to these rules:

1. Unrecognized TLVs and sub-TLVs are ignored when parsing or processing Extended-LSAs.

2. Whether or not partial deployment of a given TLV is supported MUST be specified.

3. If partial deployment is not supported, mechanisms to ensure the corresponding feature are not deployed MUST be specified in the document defining the new TLV or sub-TLV.

4. If partial deployment is supported, backward compatibility and partial deployment MUST be specified in the document defining the new TLV or sub-TLV.

5. If a TLV or Sub-TLV is recognized but the length is less than the minimum, then the LSA should be considered malformed and it SHOULD NOT be acknowledged. Additionally, the occurrence SHOULD be logged with enough information to identify the LSA by type, originator, and sequence number and the TLV or Sub-TLV in error. Ideally, the log entry would include the hexadecimal or binary representation of the LSA including the malformed TLS or Sub-TLV.

6. Documents specifying future TLVs or Sub-TLVs MUST specify the requirements for usage of those TLVs or Sub-TLVs.

7. Future TLV or Sub-TLVs must be optional. However, there may be requirements for Sub-TLVs if an optional TLV is specified.

7. Security Considerations

In general, extendible OSPFv3 LSAs are subject to the same security concerns as those described in RFC 5340 [OSPFV3]. Additionally, implementations must assure that malformed TLV and sub-TLV permutations do not result in errors that cause hard OSPFv3 failures.

If there were ever a requirement to digitally sign OSPFv3 LSAs as described for OSPFv2 LSAs in RFC 2154 [OSPF-DIGITAL-SIGNATURE], the mechanisms described herein would greatly simplify the extension.

8. IANA Considerations

This specification defines nine OSPFv3 Extended LSA types as described in Section 2. These are added the existing OSPFv3 LSA Function Codes registry.

The specification defines a new code point for the N-bit in the OSPFv3 Prefix-Options registry. The value 0x20 is suggested.

This specification also creates two registries OSPFv3 Extended-LSAs TLVs and sub-TLVs. The TLV and sub-TLV code-points in these registries are common to all Extended-LSAs and their respective definitions must define where they are applicable.

8.1. OSPFv3 Extended-LSA TLV Registry

The OSPFv3 Extended-LSA TLV registry defines top-level TLVs for Extended-LSAs and should be placed in the existing OSPFv3 IANA registry.

Nine values are allocated by this specification:

- o 0 - Reserved
- o 1 - Router-Link TLV
- o 2 - Attached-Routers TLV
- o 3 - Inter-Area Prefix TLV
- o 4 - Inter-Area Router TLV
- o 5 - External Prefix TLV

- o 6 - Intra-Area Prefix TLV
- o 7 - IPv6 Link-Local Address TLV
- o 8 - IPv4 Link-Local Address TLV

Types in the range 9-32767 are allocated via IETF Consensus or IESG Approval.

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-45055 are to be assigned on a First-Come-First-Serve (FCFS) basis.

Types in the range 45056-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

8.2. OSPFv3 Extended-LSA sub-TLV Registry

The OSPFv3 Extended-LSA sub-TLV registry defines sub-TLVs at any level of nesting for Extended-LSAs and should be placed in the existing OSPFv3 IANA registry.

Four values are allocated by this specification:

- o 0 - Reserved
- o 1 - IPv6 Forwarding Address sub-TLV
- o 2 - IPv4 Forwarding Address sub-TLV
- o 3 - Route Tag sub-TLV

Types in the range 4-32767 are allocated via IETF Consensus or IESG Approval.

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-45055 are to be assigned on a First-Come-First-Serve (FCFS) basis.

Types in the range 45056-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there

MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

9. Contributors

Contributors' Addresses

Sina Mirtorabi
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA
Email: sina@cisco.com

10. References

10.1. Normative References

[NSSA] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, January 2003.

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

[OSPFV3-AF] Lindem, A., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, April 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

[TE] Katz, D., Yeung, D., and K. Kompella, "Traffic Engineering Extensions to OSPF", RFC 3630, September 2003.

10.2. Informative References

[IPV6-ADDRESS-ARCH] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[MT-OSPFV3]
Mirtorabi, S. and A. Roy, "Multi-topology routing in OSPFv3 (MT-OSPFV3)", draft-ietf-ospf-mt-ospfv3-04.txt (work in progress), January 2008.

[OSPF-DIGITAL-SIGNATURE]
Murphy, S., Badger, M., and B. Wellington, "OSPF with Digital Signatures", RFC 2154, June 1997.

[OSPF-PREFIX-LINK]
Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPF Prefix/Link Attributes", RFC 7684, December 2015.

[SEGMENT-ROUTING]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-10.txt (work in progress), July 2016.

Appendix A. Appendix A - Global Configuration Parameters

The global configurable parameter ExtendedLSASupport is added to the OSPFv3 protocol. If ExtendedLSASupport is enabled, the OSPFv3 Router will originate OSPFv3 Extended LSAs and use the LSAs for the SPF computation. If ExtendedLSASupport is not enabled, a subset of OSPFv3 Extended LSAs may still be originated and used for other functions as described in Section 6.2.

Appendix B. Appendix B - Area Configuration Parameters

The area configurable parameter AreaExtendedLSASupport is added to the OSPFv3 protocol. If AreaExtendedLSASupport is enabled, the OSPFv3 Router will originate link and area OSPFv3 Extended LSAs and use the LSAs for the SPF computation. Legacy AS-Scoped LSAs will still be originated and used for the AS External LSA computation. If AreaExtendedLSASupport is not enabled a subset of OSPFv3 link and area Extended LSAs may still be originated and used for other functions as described in Section 6.2.

For regular areas, i.e., areas where AS scoped LSAs are flooded, disabling AreaExtendedLSASupport for a regular OSPFv3 area (not a Stub or NSSA area) when ExtendedLSASupport is enabled is contradictory and SHOULD be prohibited by the implementation.

Appendix C. Acknowledgments

OSPFv3 TLV-based LSAs were first proposed in "Multi-topology routing in OSPFv3 (MT-OSPFv3)" [MT-OSPFV3].

Thanks for Peter Psenak for significant contributions to the backward compatibility mechanisms.

Thanks go to Michael Barnes, Mike Dubrovsky, Anton Smirnov, and Tony Przygienda for review of the draft versions and discussions of backward compatibility.

Thanks to Alan Davey for review and comments including the suggestion to separate the extended LSA TLV definitions from the extended LSAs definitions.

Thanks to David Lamparter for review and suggestions on backward compatibility.

Thanks to Karsten Thomann, Chris Bowers, Meng Zhang, and Nagendra Kumar for review and editorial comments.

Thanks to Alia Atlas for substantive Routing Area Director (AD) comments prior to IETF last call.

Thanks to Alvaro Retana and Suresh Krishna for substantive comments during IESG Review.

Thanks to Mehmet Ersue for OPS Directorate review.

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

Authors’ Addresses

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

Email: acee@cisco.com

Abhay Roy
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: akr@cisco.com

Dirk Goethals
Nokia
Copernicuslaan 50
Antwerp 2018
Belgium

Email: dirk.goethals@nokia.com

Veerendranatha Reddy Vallem
Bangalore
India

Email: vallem.veerendra@gmail.com

Fred Baker
Santa Barbara, California 93117
USA

Email: FredBaker.IETF@gmail.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: 10 November 2025

Y. Qu
Futurewei Technologies
A. Lindem
LabN Consulting, L.L.C.
J. Zhang
Juniper Networks
I. Chen
The MITRE Corporation
9 May 2025

A YANG Data Model for OSPF Segment Routing over the MPLS Data Plane
draft-ietf-ospf-sr-yang-50

Abstract

This document defines a YANG data model that can be used to manage OSPF Extensions for Segment Routing over the MPLS data plane.

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 10 November 2025.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Overview 2
  1.1. Requirements Language 2
  1.2. Tree Diagrams 3
2. OSPF Segment Routing over MPLS YANG Data Model Scope 3
3. OSPF Segment Routing over MPLS YANG Module 4
4. Security Considerations 33
5. Acknowledgements 35
6. IANA Considerations 35
7. References 35
  7.1. Normative References 35
  7.2. Informative References 38
Appendix A. A Configuration Example 39
Appendix B. Full Tree Diagram 42
Authors' Addresses 52

1. Overview

This document defines a YANG data model [RFC7950] that can be used to manage OSPFv2 extensions for Segment Routing [RFC8665] and OSPFv3 extensions for Segment Routing [RFC8666] over the MPLS data plane. The defined YANG data model is an augmentation to the OSPF YANG data model [RFC9129].

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

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 model defined in [RFC8340].

2. OSPF Segment Routing over MPLS YANG Data Model Scope

The YANG data model in this document consists of a single YANG module, ietf-ospf-sr-mpls. The ietf-ospf-sr-mpls YANG module includes OSPF Segment Routing Extensions for the MPLS data plane for both OSPFv2 [RFC8665] and OSPFv3 [RFC8666].

The ietf-ospf-sr-mpls YANG module requires support for the base segment routing module [RFC9020], which defines the global segment routing configuration independent of any specific routing protocol configuration, and support of OSPF base model [RFC9129] which defines basic OSPF configuration and state.

The ietf-ospf-sr-mpls module defines both the data nodes to configure OSPF segment routing MPLS extensions and the additions to the OSPF Link State Advertisements (LSAs) necessary to support SR-MPLS. The OSPF configuration includes:

* OSPF instance level configuration imported from the ietf-segment-routing-mpls YANG module including the mapping server bindings and the per-protocol Segment Routing Global Block (SRGB) (refer to the sr-protocol grouping [RFC9020]).

* OSPF area level configuration which enables segment routing over MPLS on all interfaces as wall as advertisement of segment routing over MPLS information in Link State Advertisements (LSAs).

* OSPF interface level configuration for adjacency SIDs corresponding to specific neighbors on multi-access interfaces (OSPF interface type broadcast or Non-Broadcast Multi-Access (NBMA)). The adjacency-sid definitions are imported from tht ietf-segment-routing-mpls YANG module (refer to the igp-interface grouping [RFC9020]).

* OSPF interface level configuration for Topology-Independent Loop-Free Alternative (TI-LFA) using the MPLS data plane. TI-LFA is described in [I-D.ietf-rtgwg-segment-routing-ti-lfa].

The operational state (read-only) additions to specifc to OSPFv2 LSA include:

* OSPFv2 extended range TLV encodings [RFC8665] in the OSPF Extended-Prefix Opaque LSA [RFC7684].

* OSPFv2 Prefix SID Sub-TLV encodings [RFC8665] included the OSPF Extended Prefix TLV which is advertised in the OSPF Extended Prefix Opaque LSA [RFC7684].

* OSPFv2 Prefix SID Sub-TLV encodings [RFC8665] included the OSPF Extended Prefix TLV which is advertised in the OSPF Extended Prefix Opaque LSA [RFC7684].

* OSPFv2 SR-Algorithm, SID/Label Range TLVs, SR Local Block TLVs, Segment Routing Mapping Server (SRMS) preference TLV [RFC8665] advertised in the OSPFv2 Router Information Opaque LSA [RFC7770].

The operational state (read-only) additions to specifc to OSPFv3 LSA include:

* OSPFv3 extended range TLV encodings [RFC8666] in the OSPFv3 E-Intra-Area-Prefix-LSA, E-Inter-Area-Prefix-LSA, E-AS-External-LSA, and E-Type-7-LSA [RFC8362].

* OSPFv3 extended range TLV encodings [RFC8666] in the OSPFv3 E-Intra-Area-Prefix-LSA, E-Inter-Area-Prefix-LSA, E-AS-External-LSA, and E-Type-7-LSA [RFC8362].

* OSPFv3 Prefix-SID Sub-TLV encodings [RFC8666] in the OSPFv3 Intra-Area Prefix TLV, Inter-Area Prefix TLV, External Prefix TLV, and OSPFv3 Extended Prefix Range TLV [RFC8362].

* OSPFv3 Adj-SID Sub-TLV [RFC8666] in the OSPFv3 Router-Link TLV [RFC8362].

* OSPFv3 Adj-SID Sub-TLV [RFC8666] in the OSPFv3 Router-Link TLV [RFC8362].

* OSPFv3 LAN Adj-SID Sub-TLV [RFC8666] in the OSPFv3 Router-Link TLV [RFC8362].

* OSPFv3 SR-Algorithm, SID/Label Range TLVs, SR Local Block TLVs, Segment Routing Mapping Server (SRMS) preference TLV [RFC8666] advertised in the OSPFv3 Router Information LSA [RFC7770].

3. OSPF Segment Routing over MPLS YANG Module

[RFC2328], [RFC4915], [RFC5340], [RFC6991], [RFC8102], [RFC8294], [RFC8349], [RFC9587], and [I-D.ietf-rtgwg-segment-routing-ti-lfa] are referenced in the YANG module.

```
<CODE BEGINS> file "ietf-ospf-sr-mpls@2025-05-09.yang"
module ietf-ospf-sr-mpls {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-sr-mpls";
  prefix ospf-sr-mpls;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  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-segment-routing-common {
    prefix sr-cmn;
    reference
      "RFC 9020: YANG Data Model for Segment Routing";
  }
  import ietf-segment-routing-mpls {
    prefix sr-mpls;
    reference
      "RFC 9020: YANG Data Model for Segment Routing";
  }
  import ietf-ospf {
    prefix ospf;
    reference
      "RFC 9129: YANG Data Model for the OSPF Protocol";
  }
  import ietf-ospfv3-extended-lsa {
    prefix ospfv3-e-lsa;
    reference
      "RFC 9587: YANG Data Model for OSPFv3 Extended LSAs";
  }
```

```
     organization
       "IETF LSR - Link State Routing Working Group";
     contact
       "WG Web:   <https://datatracker.ietf.org/wg/lsr/>
        WG List:  <mailto:lsr@ietf.org>

        Author:   Yingzhen Qu
                  <mailto:yingzhen.ietf@gmail.com>
        Author:   Acee Lindem
                  <mailto:acee.ietf@gmail.com>
        Author:   Derek Yeung
                  <mailto:derek@arrcus.com>
        Author:   Jeffrey Zhang
                  <mailto:zzhang@juniper.net>
        Author:   Ing-Wher Chen
                  <mailto:ingwherchen@mitre.org>";
     description
       "This YANG module defines the generic configuration
        and operational state for OSPF Segment Routing (SR).

        Copyright (c) 2025 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 Revised BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        All revisions of IETF and IANA published modules can be found
        at the YANG Parameters registry
        (https://www.iana.org/assignments/yang-parameters).

        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 (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.";

     // RFC Ed.: Replace XXXX with actual RFC number and remove
     // this note

     revision 2025-05-09 {
       description
```

```
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for OSPF Segment Routing for
         over MPLS Data Plane";
    }

    // RFC Ed.: Replace XXXX with actual RFC number.
    // Also, replace '2025-04-14' with the module publication date
    // in the format (YYYY-MM-DD) and remove this note.

    feature remote-lfa-sr {
      description
        "Enhance rLFA to use SR path.";
      reference
        "RFC 8102: Remote-LFA Node Protection and Manageability";
    }

    feature ti-lfa {
      description
        "Topology-Independent Loop-Free Alternate (TI-LFA)
         computation using segment routing.";
      reference
        "draft-ietf-rtgwg-segment-routing-ti-lfa:
         Topology-Independent Fast Reroute using Segment Routing";
    }

    identity prefix-sid-flag {
      description
        "Base identity for prefix SID sub-TLV flags.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity np-flag {
      base prefix-sid-flag;
      description
        "No-PHP flag.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity m-flag {
      base prefix-sid-flag;
      description
        "Mapping server flag.";
      reference
```

```
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity e-flag {
      base prefix-sid-flag;
      description
        "Explicit-NULL flag.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity v-flag {
      base prefix-sid-flag;
      description
        "Value/Index flag.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity l-flag {
      base prefix-sid-flag;
      description
        "Local flag.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 6";
    }

    identity extended-prefix-range-flag {
      description
        "Base identity for extended prefix range TLV flags.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 4
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 5";
    }

    identity ia-flag {
      base extended-prefix-range-flag;
      description
        "Inter-Area flag. Note that this is only applicable to OSPFv2
         since OSPFv3 advertises separate Inter-Area extended-LSA.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 4";
    }
```

```
     identity adj-sid-flag {
       description
         "Base identity for Adj-SID sub-TLV flags.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing, Section 6
          RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
     }

     identity b-flag {
       base adj-sid-flag;
       description
         "Backup flag.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing, Section 6
          RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
     }

     identity vi-flag {
       base adj-sid-flag;
       description
         "Value/Index flag - corresponds to V-Flag in references.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing, Section 6
          RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
     }

     identity lg-flag {
       base adj-sid-flag;
       description
         "Local/Global flag - corresponds to L-flag in references.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing, Section 6
          RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
     }

     identity g-flag {
       base adj-sid-flag;
       description
         "Group flag.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing, Section 6
          RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
     }

     identity p-flag {
       base adj-sid-flag;
       description
         "Persistent flag.";
```

```
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 6
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 7";
    }

    /* Groupings */

    grouping sid-tlv-encoding {
      description
        "SID TLV Encoding -  20-bit label or 32-bit SID index whose
         interpretation is dependent on the TLV length (3 for an
         MPLS label or 4 for a 32-bit value)  or the TLV V-Flag and
         L-Flag settings:

          If the  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.

          If V-Flag is set to 1 and L-Flag is set to 1: The
          ID/Index/Label field is a 3-octet local label where the
          20 rightmost bits are used for encoding the label value.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing, Section 2.1
         RFC 8665: OSPF Extensions for Segment Routing, Section 5
         RFC 8666: OSPFv3 Extensions for Segment Routing, Section 3";
      choice sid {
        case sid-label {
          leaf label-value {
            type uint32 {
              range "0 .. 1048575";
            }
            description
              "A 20-bit MPLS Label";
          }
        }
        case sid-index {
          leaf index-value {
            type uint32;
            description
              "Index into a label space advertised by this router.";
          }
        }
        description
          "Choice of either a 20-bit MPLS label or 32-bit index into
           an advertised label space.";
      }
    }
```

```
     grouping ospfv2-prefix-sid-sub-tlvs {
       description
         "OSPFv2 Prefix Segment ID (SID) sub-TLVs.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing
           (Section 5)";
       container prefix-sid-sub-tlvs {
         description
           "Prefix SID sub-TLV.";
         list prefix-sid-sub-tlv {
           description
             "Prefix SID sub-TLV.";
           container prefix-sid-flags {
             leaf-list flag {
               type identityref {
                 base prefix-sid-flag;
               }
               description
                 "Prefix SID Sub-TLV flags.";
             }
             description
               "Segment Identifier (SID) Flags.";
           }
           leaf mt-id {
             type uint8 {
               range "0 .. 127";
             }
             description
               "Multi-topology ID.";
             reference
               "RFC 4915: Multi-Topology (MT) Routing in OSPF";
           }
           leaf algorithm {
             type identityref {
               base sr-cmn:prefix-sid-algorithm;
             }
             description
               "Algorithm associated with the prefix-SID.";
           }
           uses sid-tlv-encoding;
         }
       }
     }

     grouping ospfv2-extended-prefix-range-tlvs {
       description
         "OSPFv2 Extended prefix range TLV grouping.";
       reference
```

```
         "RFC 8665: OSPF Extensions for Segment Routing
          (Section 4)";
       container extended-prefix-range-tlvs {
         description
           "List of range of prefixes.";
         list extended-prefix-range-tlv {
           description
             "Range of prefixes.";
           leaf prefix-length {
             type uint8;
             description
               "Length of prefix in bits.";
           }
           leaf af {
             type iana-rt-types:address-family;
             description
               "Address family for the prefix.";
           }
           leaf range-size {
             type uint16;
             description
               "Number of prefixes covered by the
                advertisement.";
           }
           container extended-prefix-range-flags {
             leaf-list flag {
               type identityref {
                 base extended-prefix-range-flag;
               }
               description
                 "Extended prefix range TLV flags.";
             }
             description
               "Extended Prefix Range TLV flags.";
           }
           leaf prefix {
             type inet:ipv4-prefix;
             description
               "IPv4 prefix.";
           }
           uses ospfv2-prefix-sid-sub-tlvs;
           uses ospf:unknown-tlvs;
         }
       }
     }

     grouping ospfv2-adj-sid-sub-tlvs {
       description
```

```
        "OSPFv2 Adj-SID sub-TLV grouping.";
      container adj-sid-sub-tlvs {
        description
          "Adj-SID optional sub-TLVs.";
        list adj-sid-sub-tlv {
          description
            "List of Adj-SID sub-TLVs.";
          container adj-sid-flags {
            leaf-list flag {
              type identityref {
                base adj-sid-flag;
              }
              description
                "Adj-SID sub-TLV flags.";
            }
            description
              "Adj-SID sub-TLV flags.";
          }
          leaf mt-id {
            type uint8 {
              range "0 .. 127";
            }
            description
              "Multi-topology ID. Topologies range from 0-127 and
               return of any other value would indicate an error.";
            reference
              "RFC 4915: Multi-Topology (MT) Routing in OSPF";
          }
          leaf weight {
            type uint8;
            description
              "Weight used for load-balancing.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    grouping ospfv2-lan-adj-sid-sub-tlvs {
      description
        "OSPFv2 LAN Adj-SID sub-TLV grouping.";
      container lan-adj-sid-sub-tlvs {
        description
          "LAN Adj-SID optional sub-TLVs.";
        reference
          "RFC 8665: OSPF Extensions for Segment Routing
           (Section 6)";
        list lan-adj-sid-sub-tlv {
```

```
          description
            "List of LAN Adj-SID sub-TLVs.";
          container lan-adj-sid-flags {
            leaf-list flag {
              type identityref {
                base adj-sid-flag;
              }
              description
                "LAN Adj-SID sub-TLV flags.";
            }
            description
              "LAN Adj-SID sub-TLV flags.";
          }
          leaf mt-id {
            type uint8 {
              range "0 .. 127";
            }
            description
              "Multi-topology ID. Topologies range from 0-127 and
               return of any other value would indicate an error.";
            reference
              "RFC 4915: Multi-Topology (MT) Routing in OSPF";
          }
          leaf weight {
            type uint8;
            description
              "Weight used for load-balancing.";
          }
          leaf neighbor-router-id {
            type rt-types:router-id;
            description
              "Neighbor router ID.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    grouping sr-algorithm-tlv {
      description
        "SR algorithm TLV grouping.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
         (Section 3.1)";
      container sr-algorithm-tlv {
        description
          "All SR algorithm TLVs.";
        leaf-list sr-algorithm {
```

```
          type identityref {
            base sr-cmn:prefix-sid-algorithm;
          }
          description
            "Segment Routing (SR) algorithms that the router is
             currently using.";
        }
      }
    }

    grouping sid-range-tlvs {
      description
        "SID Range TLV grouping.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
         (Section 3.2)";
      container sid-range-tlvs {
        description
          "List of SID range TLVs.";
        list sid-range-tlv {
          description
            "SID range TLV.";
          leaf range-size {
            type rt-types:uint24;
            description
              "SID range.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    grouping local-block-tlvs {
      description
        "The SR local block TLV contains the
         range of labels reserved for local SIDs.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
         (Section 3.3)";
      container local-block-tlvs {
        description
          "List of SRLB TLVs.";
        list local-block-tlv {
          description
            "SRLB TLV.";
          leaf range-size {
            type rt-types:uint24;
            description
```

```
               "SID range. The return of a zero value would indicate
                an error.";
           }
           uses sid-tlv-encoding;
         }
       }
     }

     grouping srms-preference-tlv {
       description
         "The SR Mapping Server (SRMS) preference TLV is
          used to advertise a preference associated with
          the node that acts as an SR Mapping Server. SR
          Mapping Server advertisements with a higher
          preference value are preferred over those with
          a lower preference value.";
       reference
         "RFC 8665: OSPF Extensions for Segment Routing
          (Section 3.4)";
       container srms-preference-tlv {
         description
           "SRMS Preference TLV.";
         leaf preference {
           type uint8;
           description
             "SRMS preference TLV, value from 0 to 255 with
              255 being the most preferred.";
         }
       }
     }

     grouping ospfv3-prefix-sid-sub-tlvs {
       description
         "OSPFv3 Prefix Segment ID (SID) sub-TLVs.";
       reference
         "RFC 8666: OSPFv3 Extensions for Segment Routing
          (Section 6)";
       container prefix-sid-sub-tlvs {
         description
           "Prefix SID sub-TLV.";
         list prefix-sid-sub-tlv {
           description
             "Prefix SID sub-TLV.";
           container ospfv3-prefix-sid-flags {
             leaf-list flag {
               type identityref {
                 base prefix-sid-flag;
               }
```

```
              description
                "Prefix SID sub-TLV flags.";
            }
            description
              "Segment Identifier (SID) Flags.";
          }
          leaf algorithm {
            type identityref {
              base sr-cmn:prefix-sid-algorithm;
            }
            description
              "Algorithm associated with the prefix-SID.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    grouping ospfv3-extended-prefix-range-tlvs {
      description
        "OSPFv3 Extended prefix range TLV grouping.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 5)";
      container ospfv3-extended-prefix-range-tlvs {
        description
          "List of extended prefix range TLVs.";
        list extended-prefix-range-tlv {
          description
            "Range of prefixes.";
          leaf prefix-length {
            type uint8;
            description
              "Length of prefix in bits.";
          }
          leaf af {
            type iana-rt-types:address-family;
            description
              "Address family for the prefix.";
          }
          leaf range-size {
            type uint16;
            description
              "Number of prefixes covered by the advertisement.
               The return of a value of zero would indicate an error.";
          }
          leaf prefix {
            type inet:ip-prefix;
```

```
            description
              "IPv4 or IPv6 prefix.";
          }
          uses ospfv3-prefix-sid-sub-tlvs;
          uses ospf:unknown-tlvs;
        }
      }
    }

    grouping ospfv3-adj-sid-sub-tlvs {
      description
        "OSPFv3 Adj-SID sub-TLV grouping.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 7)";
      container adj-sid-sub-tlvs {
        description
          "Adj-SID optional sub-TLVs.";
        list adj-sid-sub-tlv {
          description
            "List of Adj-SID sub-TLVs.";
          container adj-sid-flags {
            leaf-list flag {
              type identityref {
                base adj-sid-flag;
              }
              description
                "Adj-SID sub-tlv flags.";
            }
            description
              "Adj-sid sub-tlv flags.";
          }
          leaf weight {
            type uint8;
            description
              "Weight used for load-balancing.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    grouping ospfv3-lan-adj-sid-sub-tlvs {
      description
        "OSPFv3 LAN Adj-SID sub-TLV grouping.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 7)";
```

```
      container lan-adj-sid-sub-tlvs {
        description
          "LAN Adj-SID optional sub-TLVs.";
        list lan-adj-sid-sub-tlv {
          description
            "List of LAN Adj-SID sub-TLVs.";
          container lan-adj-sid-flags {
            leaf-list flag {
              type identityref {
                base adj-sid-flag;
              }
              description
                "LAN Adj-SID sub-TLV flags.";
            }
            description
              "LAN Adj-SID sub-TLV flags.";
          }
          leaf weight {
            type uint8;
            description
              "Weight used for load-balancing.";
          }
          leaf neighbor-router-id {
            type rt-types:router-id;
            description
              "Neighbor router ID.";
          }
          uses sid-tlv-encoding;
        }
      }
    }

    /*
     * Augmentations for OSPF SR MPLS Configuration (read-write)
     * data nodes.
     */

    augment "/rt:routing/rt:control-plane-protocols"
          + "/rt:control-plane-protocol/ospf:ospf" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf')" {
        description
          "This augments the OSPF routing protocol when used.";
      }
      description
        "This augments the OSPF protocol configuration with segment
         routing over the MPLS data plane. The following semantic
         validation be performed for the configuration data:
```

```
          - Assure the binding policies prefixes do not overlapp.";
      reference
        "RFC 9020 - YANG Data Model for Segment Routing";
      uses sr-mpls:sr-control-plane;
      container protocol-srgb {
        if-feature "sr-mpls:protocol-srgb";
        uses sr-cmn:srgb;
        description
          "Per-protocol SRGB.";
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/"
          + "ospf:areas/ospf:area" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf') and "
         + "/rt:routing/rt:control-plane-protocols"
         + "/rt:control-plane-protocol/ospf:ospf"
         + "/ospf-sr-mpls:segment-routing/"
         + "ospf-sr-mpls:enabled = 'true'" {
        description
          "This augments the OSPF area configuration when segment
           routing is enabled at the OSPF instance level.";
      }
      container segment-routing {
        presence "When present segment routing over MPLS is enabled
                  for the area.";
        description
          "OSPF area level segment routing configuration. Enable
           segment routing over MPLS on all interfaces and enables
           advertisement of Link State Advertisements (LSAs) and TLVs
           supporting segment routing over MPLS";
      }
      description
        "This augments the OSPF protocol area configuration with
         segment routing.";
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/"
          + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf') and "
         + "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/ospf:ospf/"
         + "ospf:areas/ospf:area/ospf-sr-mpls:segment-routing" {
        description
```

```
          "This augments the OSPF interface configuration when used.";
      }
      description
        "This augments the OSPF protocol interface
         configuration with segment routing.";
      reference
        "RFC 9020 - YANG Data Model for Segment Routing";
      uses sr-mpls:igp-interface {
        augment "segment-routing/adjacency-sid/adj-sids" {
          when "((../../../ospf:interface-type = 'broadcast') or
                (../../../ospf:interface-type = 'non-broadcast'))" {
            description
              "This augments broadcast and non-broadcast multi-access
               interface segment routing interface configuration.";
          }
          description
            "This augments LAN interface adj-sid with neighbor-id.";
          leaf neighbor-id {
            type inet:ip-address;
            mandatory true;
            description
              "Neighbor's Router ID, IPv4 address, or IPv6 address.
               Specification is optional and, if specified, SHOULD
               specify a neighbor reachable via the interface.";
          }
        }
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/"
          + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
          + "ospf:fast-reroute/ospf:lfa" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf')" {
        description
          "This augments the OSPF routing protocol when used.";
      }
      description
        "This augments the OSPF protocol IP-FRR with TI-LFA.";
      container ti-lfa {
        if-feature "ti-lfa";
        description
          "Topology-Independent Loop Free Alternate
           (TI-LFA) support.";
        reference
          "draft-ietf-rtgwg-segment-routing-ti-lfa:
           Topology-Independent Fast Reroute using Segment Routing";
```

```
        leaf enabled {
          type boolean;
          default "false";
          description
            "Enable TI-LFA computation.";
        }
        container selection-tie-breakers {
          container node-protection {
            presence "Presence of container enables the node
                      protection tie-breaker.";
            leaf priority {
              type uint8;
              default "128";
              description
                "Priority for node protection tie-breaker with
                 a lower priority being more preferred. By default,
                 the priority is in the middle of the priority range.";
            }
            description
              "Enable node protection as a TI-LFA path
               selection tie-breaker. A path providing node
               protection will be selected over one that
               doesn't provide node protection.";
          }
          container srlg-disjoint {
            presence "Presence of container enables the SRLG
                      disjoint tie-breaker";
            leaf priority {
              type uint8;
              default "128";
              description
                "Priority for SRLG disjoint tie-breaker with
                 a lower priority being more preferred. By default,
                 the priority is in the middle of the priority range.";
            }
            description
              "Enable SRLG (Sharsed Resource Link Group)
               disjoint as a TI-LFA path selection tie-breaker.
               A path providing node a disjoint path for SRLG
               links from the primary path will be selected over
               one that doesn't provide an SRLG disjoint path.";
          }
          description
            "Configure path selection tie-breakers and their
             respective priorities for the TI-LFA computation.
             multiple tie-breakers and priorities configured may
             be configured.";
        }
```

```
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/"
          + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
          + "ospf:fast-reroute/ospf:lfa/ospf:remote-lfa" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf')" {
        description
          "This augments the OSPF routing protocol when used.";
      }
      description
        "This augments the OSPF protocol IP-FRR with remote LFA.";
      reference
        "draft-ietf-rtgwg-segment-routing-ti-lfa:
         Topology-Independent Fast Reroute using Segment Routing";
      leaf use-segment-routing-path {
        if-feature "remote-lfa-sr";
        type boolean;
        default "false";
        description
          "Force remote LFA to use segment routing path instead of LDP
           path. The value of this leaf is in effect only when
           remote-lfa is enabled.";
      }
    }

    /* Operational states */

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/ospf:ospf/"
          + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
          + "ospf:neighbors/ospf:neighbor" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospf')" {
        description
          "This augments the OSPF routing protocol when used.";
      }
      description
        "This augments OSPF interface adjancency-sid state.";
      list adjacency-sid {
        description
          "List of adjacency Segment IDs.";
        leaf value {
          type uint32;
          description
            "Value of the Adj-SID.";
```

```
        }
        leaf weight {
          type uint8;
          description
            "Weight associated with the adjacency SID.";
        }
        leaf protection-requested {
          type boolean;
          description
            "Describe if the adjacency SID is protected.";
        }
      }
    }

    /*
     * Augmentations for read-only data nodes corresponding to Segment
     * Routing encodings in OSPFv2 and OSPFv3 Link State Advertisements
     * (LSAs) in OSPF Link State Databases (LSDBs) at the instance,
     * area, and interface level.
     */
    /* Augmentations for OSPFv2 LSAs. */

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/"
          + "ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/"
          + "ospf:extended-prefix-opaque" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR Extended Prefix Range TLV in OSPFv2 Type 10 (area-scoped)
         Extended Prefix Opaque LSAs.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
         (Section 4)";
      uses ospfv2-extended-prefix-range-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:database/"
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
```

```
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/"
          + "ospf:extended-prefix-opaque" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR Extended Prefix Range TLV in OSPFv2 Type 11 (AS-scoped)
         Extended Prefix Opaque LSAs.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
         (Section 4)";
      uses ospfv2-extended-prefix-range-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/"
          + "ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/"
          + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR Prefix-SID sub-TLVs for OSPFv2 Extended Prefix TLVs
         in OSPFv2 Type 10 (area-scoped) Extended Prefix Opaque LSAs.";
      uses ospfv2-prefix-sid-sub-tlvs;
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
          (Section 5)";
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:database/"
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/"
          + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
```

```
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR Prefix-SID sub-TLVs for OSPFv2 Extended Prefix TLVs
         in OSPFv2 Type 11 (AS-scoped) Extended Prefix Opaque LSAs.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
          (Section 5)";
      uses ospfv2-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/"
          + "ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/"
          + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR TLVs for OSPFv2 Extended Link TLV in OSPFv2 Type 10
         (area-scoped) Extended Link Opaque LSAs.";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
          (Section 6)";
      uses ospfv2-adj-sid-sub-tlvs;
      uses ospfv2-lan-adj-sid-sub-tlvs;
    }

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

```
        "SR TLVs for OSPFv2 Router Information Type 10 (area-scoped)
         Opaque LSAs";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
          (Section 3)";
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:database/"
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
          + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
        description
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR TLVs for OSPFv2 Router Information Type 11 (AS-scoped)
         Opaque LSAs";
      reference
        "RFC 8665: OSPF Extensions for Segment Routing
          (Section 3)";
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }

    /* Augmentations for OSPFv3 LSAs. */

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:database/"
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospf:router-information" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
```

```
        "SR-specific TLVs for OSPFv3 Router Information LSA
         (area-scoped).";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 4)";
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/"
          + "ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospf:router-information" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR-specific TLVs for OSPFv3 Router Information LSA
         (AS-scoped).";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 4)";
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-intra-area-prefix" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      uses ospfv3-extended-prefix-range-tlvs;
      description
```

```
         "SR Extended Prefix Range TLVs in OSPFv3
          E-Intra-Area-Prefix LSAs.";
     }

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
           + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
           + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
           + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-inter-area-prefix" {
       when "derived-from(/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
         description
           "This augmentation is only valid for OSPFv3.";
       }
       description
         "SR Extended Prefix Range TLVs in OSPFv3
          E-Inter-Area-Prefix LSAs.";
       reference
         "RFC 8666: OSPFv3 Extensions for Segment Routing
          (Section 5)";
       uses ospfv3-extended-prefix-range-tlvs;
     }

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:database/"
           + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
           + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
           + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-as-external" {
       when "derived-from(/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
         description
           "This augmentation is only valid for OSPFv3.";
       }
       description
         "SR Extended Prefix Range TLVs in OSPFv3 E-AS-External LSAs.";
       reference
         "RFC 8666: OSPFv3 Extensions for Segment Routing
          (Section 5)";
       uses ospfv3-extended-prefix-range-tlvs;
     }

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
           + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
           + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
```

```
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-nssa" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR Extended Prefix Range TLVs in OSPFv3 E-NSSA LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 5)";
      uses ospfv3-extended-prefix-range-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:interfaces/"
          + "ospf:interface/"
          + "ospf:database/ospf:link-scope-lsa-type/"
          + "ospf:link-scope-lsas/ospf:link-scope-lsa/"
          + "ospf:version/ospf:ospfv3/ospf:ospfv3/"
          + "ospf:body/ospfv3-e-lsa:e-link/"
          + "ospfv3-e-lsa:e-link-tlvs/ospfv3-e-lsa:intra-prefix-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR Prefix-SID Sub-TLV in OSPFv3 Intra-Area
         Prefix TLV for OSPFv3 E-Link LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 6)";
      uses ospfv3-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-intra-area-prefix/"
          + "ospfv3-e-lsa:e-intra-prefix-tlvs/"
          + "ospfv3-e-lsa:intra-prefix-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
```

```
          + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3
           E-Router LSAs";
      }
      description
        "SR Prefix-SID Sub-TLV in OSPFv3 Intra-Area
         Prefix TLV for OSPFv3 E-Intra-Area-Prefix LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 6)";
      uses ospfv3-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-inter-area-prefix/"
          + "ospfv3-e-lsa:e-inter-prefix-tlvs/"
          + "ospfv3-e-lsa:inter-prefix-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR Prefix-SID Sub-TLV in OSPFv3 Link-Scoped Intra-Area
         Prefix TLV for OSPFv3 E-Inter-Area-Prefix LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 6)";
      uses ospfv3-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:database/"
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-as-external/"
          + "ospfv3-e-lsa:e-external-tlvs/"
          + "ospfv3-e-lsa:external-prefix-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
```

```
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR Prefix-SID Sub-TLV in OSPFv3 External Prefix TLV
         for OSPFv3 E-AS-External LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 6)";
      uses ospfv3-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-nssa/"
          + "ospfv3-e-lsa:e-external-tlvs/"
          + "ospfv3-e-lsa:external-prefix-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3.";
      }
      description
        "SR Prefix-SID Sub-TLV in OSPFv3 External Prefix TLV
         for OSPFv3 E-NSSA LSAs.";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 6)";
      uses ospfv3-prefix-sid-sub-tlvs;
    }

    augment "/rt:routing/"
          + "rt:control-plane-protocols/rt:control-plane-protocol/"
          + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
          + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
          + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router/"
          + "ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv/"
          + "ospfv3-e-lsa:sub-tlvs" {
      when "derived-from(/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
        description
          "This augmentation is only valid for OSPFv3
           E-Router LSAs";
```

```
      }
      description
        "SR Sub-TLVs in OSPFv3 link-tlv for OSPFv3 E-Router LSAs";
      reference
        "RFC 8666: OSPFv3 Extensions for Segment Routing
         (Section 7)";
      uses ospfv3-adj-sid-sub-tlvs;
      uses ospfv3-lan-adj-sid-sub-tlvs;
    }
  }
  <CODE ENDS>
```

Figure 1

4. Security Considerations

The ietf-ospf-sr-mpls YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be reasonably sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

/ospf:ospf/segment-routing/enabled - Modification to the enablement for SR could result in a Denial-of-Service (DoS) attack. If an attacker disables SR, it will cause traffic disruption.

/ospf:ospf/segment-routing/bindings - Modification to the local bindings could result in a Denial-of-Service (DoS) attack.

/ospf:ospf/protocol-srgb - Modification of the protocol SRGB could be used to mount a DoS attack. For example, if the protocol SRBG size is reduced to a very small value, a lot of existing segments could no longer be installed leading to a traffic disruption.

/ospf:interfaces/ospf:interface/segment-routing - Modification of the Adjacency Segment Identifier (Adj-SID) could be used to mount a DoS attack. Change of an Adj-SID could be used to redirect traffic.

/ospf:interfaces/ospf:interface/ospf:fast-reroute/ti-lfa - Modification of the TI-LFA enablement could lead to traffic disruption.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/ vulnerabilities:

The module ietf-ospf-sr-mpls augments base OSPF module Link State Database (LSDB) with various TLVs. Knowledge of these data nodes can be used to attack other routers in the OSPF domain. These attacks are documented in [RFC9129].

Knowledge of the Prefix and Adjacency SIDs advertised in LSAs could facilitate a targeted attack on the data-plane by knowing the topology and path for MPLS segment-routed packets. Examples include the OSPFv2 extended-prefix-opaque-LSA/prefix-sid-sub-tlvs and OSPFv3 e-router-tlvs/link-tlv/sub-tlvs/ospfv3-lan-adj-sid-sub-tlvs.

Knowledge of Prefix and Adjaceny SIDs may also divulge a view of the segment routing network topology architecture which may be considered a proprietary asset.

Knowledge of the advertised SID/Label Range TLVs, SR Local Block TLVs and Segment Routing Mapping Server (SRMS) TLVs may facilitate a Denial of Service (DoS) attack on the advertising node’s SR and SRMS functionality. Examples include augmentations to the OSPFv2 ri-opaque/sid-range-tlvs and OSPFv3 router-information/local-block-tlvs.

This YANG module uses groupings from other YANG modules that define nodes that may be considered sensitive or vulnerable in network environments. Refer to the Security Considerations of [RFC9020] for information as to which nodes may be considered sensitive or vulnerable in network environments.

5. Acknowledgements

The authors wish to thank Dean Bogdanovic and Kiran Koushik Agrahara Sreenivasa for their YANG module discussions.

The authors wish to thank Yi Yang, Alexander Clemm, Gaurav Gupta, Ladislav Lhotka, Stephane Litkowski, Greg Hankins, Manish Gupta, Alan Davey, Divakaran Baskaran, Reshad Rahman, Tom Petch, and Corey Bonnell for their thorough reviews and helpful comments.

The authors wish to thank Julien Meuric for a very meticulous Routing Directorate review.

The authors wish to thank Mohamed Boucadair and Ketan Talaulikar for extensive AD reviews for the IESG Telechat. Thanks to Eric Vyncke and Mahesh Jethanandani for review comments.

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 agreed to this document for Public Release, Distribution Unlimited, with Public Release Case Number 18-3281. This approval allows the MITRE employee to release contributions externally and does not assert authority to publish.

6. 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:

```
URI: urn:ietf:params:xml:ns:yang:ietf-ospf-sr-mpls
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-sr-mpls
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-sr-mpls
prefix: ospf-sr-mpls
maintained by IANA? N
reference: RFC XXXX
```

7. References

7.1. Normative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Litkowski, S., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-21, 12 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-21>.

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

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

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

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

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

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

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

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

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

[RFC8102] Sarkar, P., Ed., Hegde, S., Bowers, C., Gredler, H., and S. Litkowski, "Remote-LFA Node Protection and Manageability", RFC 8102, DOI 10.17487/RFC8102, March 2017, <https://www.rfc-editor.org/info/rfc8102>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

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

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

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

[RFC9020] Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", RFC 9020, DOI 10.17487/RFC9020, May 2021, <https://www.rfc-editor.org/info/rfc9020>.

[RFC9129] Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for the OSPF Protocol", RFC 9129, DOI 10.17487/RFC9129, October 2022, <https://www.rfc-editor.org/info/rfc9129>.

[RFC9587] Lindem, A., Palani, S., and Y. Qu, "YANG Data Model for OSPFv3 Extended Link State Advertisements (LSAs)", RFC 9587, DOI 10.17487/RFC9587, June 2024, <https://www.rfc-editor.org/info/rfc9587>.

7.2. Informative References

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

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

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Appendix A. A Configuration Example

The following is an XML example using the ietf-ospf-sr-mpls YANG module, and RFC 9020.

Note: '\' line wrapping per [RFC8792].

```
<?xml version='1.0' encoding='UTF-8'?>
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <router-id>1.1.1.1</router-id>
    <control-plane-protocols>
      <control-plane-protocol>
        <type xmlns:ospf="urn:ietf:params:xml:ns:yang:ietf-ospf">\
          ospf:ospfv2</type>
        <name>OSPFv2</name>
        <ospf xmlns="urn:ietf:params:xml:ns:yang:ietf-ospf">
          <areas>
            <area>
              <area-id>0.0.0.0</area-id>
              <segment-routing xmlns="urn:ietf:params:xml:ns:yang:\
                ietf-ospf-sr-mpls"></segment-routing>
              <interfaces>
                <interface>
                  <name>eth0</name>
                  <segment-routing xmlns="urn:ietf:params:xml:ns:\
                    yang:ietf-ospf-sr-mpls">
                    <adjacency-sid>
                      <adj-sids>
                        <value>3888</value>
                      </adj-sids>
                    </adjacency-sid>
                  </segment-routing>
                </interface>
              </interfaces>
            </area>
          </areas>
          <segment-routing xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-ospf-sr-mpls">
            <enabled>true</enabled>
          </segment-routing>
          <protocol-srgb xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-ospf-sr-mpls">
            <srgb>
              <lower-bound>4000</lower-bound>
              <upper-bound>5000</upper-bound>
            </srgb>
          </protocol-srgb>
        </ospf>
      </control-plane-protocol>
    </control-plane-protocols>
  </routing>
```

The following is the same example using JSON format.

```
{
  "ietf-routing:routing": {
    "router-id": "1.1.1.1",
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-ospf:ospfv2",
          "name": "OSPFv2",
          "ietf-ospf:ospf": {
            "areas": {
              "area": [
                {
                  "area-id": "0.0.0.0",
                  "interfaces": {
                    "interface": [
                      {
                        "name": "eth0",
                        "ietf-ospf-sr-mpls:segment-routing": {
                          "adjacency-sid": {
                            "adj-sids": [
                              {
                                "value": 3888
                              }
                            ]
                          }
                        }
                      }
                    ]
                  },
                  "ietf-ospf-sr-mpls:segment-routing": {}
                }
              ]
            },
            "ietf-ospf-sr-mpls:segment-routing": {
              "enabled": true
            },
            "ietf-ospf-sr-mpls:protocol-srgb": {
              "srgb": [
                {
                  "lower-bound": 4000,
                  "upper-bound": 5000
                }
              ]
            }
          }
        }
      ]
    }
```

```
     }
   }
```

Appendix B. Full Tree Diagram

```
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf:
     +--rw segment-routing
     |  +--rw enabled?    boolean
     |  +--rw bindings {mapping-server}?
     |     +--rw advertise
     |     |  +--rw policies*   leafref
     |     +--rw receive?     boolean
     +--rw protocol-srgb {sr-mpls:protocol-srgb}?
        +--rw srgb* [lower-bound upper-bound]
           +--rw lower-bound    uint32
           +--rw upper-bound    uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area:
     +--rw segment-routing!
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:interfaces/ospf:interface:
     +--rw segment-routing
        +--rw adjacency-sid
           +--rw adj-sids* [value]
           |  +--rw value-type?    enumeration
           |  +--rw value          uint32
           |  +--rw protected?     boolean
           |  +--rw weight?        uint8
           |  +--rw neighbor-id    inet:ip-address
           +--rw advertise-adj-group-sid* [group-id]
           |  +--rw group-id    uint32
           +--rw advertise-protection?      enumeration
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:interfaces/ospf:interface
             /ospf:fast-reroute/ospf:lfa:
     +--rw ti-lfa {ti-lfa}?
        +--rw enabled?                  boolean
        +--rw selection-tie-breakers
           +--rw node-protection!
           |  +--rw priority?   uint8
           +--rw srlg-disjoint!
              +--rw priority?   uint8
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
```

```
            /ospf:area/ospf:interfaces/ospf:interface
            /ospf:fast-reroute/ospf:lfa/ospf:remote-lfa:
     +--rw use-segment-routing-path?   boolean {remote-lfa-sr}?
   augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:areas
            /ospf:area/ospf:interfaces/ospf:interface/ospf:neighbors
            /ospf:neighbor:
     +--ro adjacency-sid* []
        +--ro value?                  uint32
        +--ro weight?                 uint8
        +--ro protection-requested?   boolean
   augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:areas
            /ospf:area/ospf:database/ospf:area-scope-lsa-type
            /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
            /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
            /ospf:extended-prefix-opaque:
     +--ro extended-prefix-range-tlvs
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?                 uint8
           +--ro af?
           |       iana-rt-types:address-family
           +--ro range-size?                    uint16
           +--ro extended-prefix-range-flags
           |  +--ro flag*   identityref
           +--ro prefix?                        inet:ipv4-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro mt-id?              uint8
           |     +--ro algorithm?          identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?   uint32
           |        +--:(sid-index)
           |           +--ro index-value?   uint32
           +--ro unknown-tlvs
              +--ro unknown-tlv* []
                 +--ro type?     uint16
                 +--ro length?   uint16
                 +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:database
            /ospf:as-scope-lsa-type/ospf:as-scope-lsas
            /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
            /ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
     +--ro extended-prefix-range-tlvs
```

```
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?                 uint8
           +--ro af?
           |       iana-rt-types:address-family
           +--ro range-size?                    uint16
           +--ro extended-prefix-range-flags
           |  +--ro flag*   identityref
           +--ro prefix?                        inet:ipv4-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro mt-id?               uint8
           |     +--ro algorithm?           identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?   uint32
           |        +--:(sid-index)
           |           +--ro index-value?   uint32
           +--ro unknown-tlvs
              +--ro unknown-tlv* []
                 +--ro type?     uint16
                 +--ro length?   uint16
                 +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
             /ospf:extended-prefix-opaque/ospf:extended-prefix-tlv:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
           +--ro prefix-sid-flags
           |  +--ro flag*   identityref
           +--ro mt-id?               uint8
           +--ro algorithm?           identityref
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?   uint32
              +--:(sid-index)
                 +--ro index-value?   uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:database
             /ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
             /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
             /ospf:extended-prefix-tlv:
     +--ro prefix-sid-sub-tlvs
```

```
        +--ro prefix-sid-sub-tlv* []
           +--ro prefix-sid-flags
           |  +--ro flag*   identityref
           +--ro mt-id?               uint8
           +--ro algorithm?           identityref
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?   uint32
              +--:(sid-index)
                 +--ro index-value?   uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
             /ospf:extended-link-opaque/ospf:extended-link-tlv:
     +--ro adj-sid-sub-tlvs
     |  +--ro adj-sid-sub-tlv* []
     |     +--ro adj-sid-flags
     |     |  +--ro flag*   identityref
     |     +--ro mt-id?               uint8
     |     +--ro weight?              uint8
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro lan-adj-sid-sub-tlvs
        +--ro lan-adj-sid-sub-tlv* []
           +--ro lan-adj-sid-flags
           |  +--ro flag*   identityref
           +--ro mt-id?                uint8
           +--ro weight?               uint8
           +--ro neighbor-router-id?   rt-types:router-id
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?    uint32
              +--:(sid-index)
                 +--ro index-value?    uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
             /ospf:ri-opaque:
     +--ro sr-algorithm-tlv
     |  +--ro sr-algorithm*   identityref
     +--ro sid-range-tlvs
```

```
     |  +--ro sid-range-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro local-block-tlvs
     |  +--ro local-block-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro srms-preference-tlv
        +--ro preference?   uint8
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:database
             /ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
             /ospf:body/ospf:opaque/ospf:ri-opaque:
     +--ro sr-algorithm-tlv
     |  +--ro sr-algorithm*   identityref
     +--ro sid-range-tlvs
     |  +--ro sid-range-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro local-block-tlvs
     |  +--ro local-block-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro srms-preference-tlv
        +--ro preference?   uint8
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:database
             /ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
             /ospf:body/ospf:router-information:
     +--ro sr-algorithm-tlv
```

```
     |  +--ro sr-algorithm*   identityref
     +--ro sid-range-tlvs
     |  +--ro sid-range-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro local-block-tlvs
     |  +--ro local-block-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro srms-preference-tlv
        +--ro preference?   uint8
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body
             /ospf:router-information:
     +--ro sr-algorithm-tlv
     |  +--ro sr-algorithm*   identityref
     +--ro sid-range-tlvs
     |  +--ro sid-range-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro local-block-tlvs
     |  +--ro local-block-tlv* []
     |     +--ro range-size?          rt-types:uint24
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro srms-preference-tlv
        +--ro preference?   uint8
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
```

```
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body
             /ospfv3-e-lsa:e-intra-area-prefix:
     +--ro ospfv3-extended-prefix-range-tlvs
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?         uint8
           +--ro af?                    iana-rt-types:address-family
           +--ro range-size?            uint16
           +--ro prefix?                inet:ip-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro ospfv3-prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro algorithm?                 identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?         uint32
           |        +--:(sid-index)
           |           +--ro index-value?         uint32
           +--ro unknown-tlvs
              +--ro unknown-tlv* []
                 +--ro type?     uint16
                 +--ro length?   uint16
                 +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body
             /ospfv3-e-lsa:e-inter-area-prefix:
     +--ro ospfv3-extended-prefix-range-tlvs
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?         uint8
           +--ro af?                    iana-rt-types:address-family
           +--ro range-size?            uint16
           +--ro prefix?                inet:ip-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro ospfv3-prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro algorithm?                 identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?         uint32
           |        +--:(sid-index)
           |           +--ro index-value?         uint32
           +--ro unknown-tlvs
              +--ro unknown-tlv* []
```

```
                +--ro type?     uint16
                +--ro length?   uint16
                +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:database
             /ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
             /ospf:body/ospfv3-e-lsa:e-as-external:
     +--ro ospfv3-extended-prefix-range-tlvs
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?         uint8
           +--ro af?                    iana-rt-types:address-family
           +--ro range-size?            uint16
           +--ro prefix?                inet:ip-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro ospfv3-prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro algorithm?                identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?        uint32
           |        +--:(sid-index)
           |           +--ro index-value?        uint32
           +--ro unknown-tlvs
              +--ro unknown-tlv* []
                 +--ro type?     uint16
                 +--ro length?   uint16
                 +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-nssa:
     +--ro ospfv3-extended-prefix-range-tlvs
        +--ro extended-prefix-range-tlv* []
           +--ro prefix-length?         uint8
           +--ro af?                    iana-rt-types:address-family
           +--ro range-size?            uint16
           +--ro prefix?                inet:ip-prefix
           +--ro prefix-sid-sub-tlvs
           |  +--ro prefix-sid-sub-tlv* []
           |     +--ro ospfv3-prefix-sid-flags
           |     |  +--ro flag*   identityref
           |     +--ro algorithm?                identityref
           |     +--ro (sid)?
           |        +--:(sid-label)
           |        |  +--ro label-value?        uint32
```

```
          |        +--:(sid-index)
          |           +--ro index-value?         uint32
          +--ro unknown-tlvs
             +--ro unknown-tlv* []
                +--ro type?     uint16
                +--ro length?   uint16
                +--ro value?    yang:hex-string
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:interfaces/ospf:interface/ospf:database
             /ospf:link-scope-lsa-type/ospf:link-scope-lsas
             /ospf:link-scope-lsa/ospf:version/ospf:ospfv3
             /ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-link
             /ospfv3-e-lsa:e-link-tlvs/ospfv3-e-lsa:intra-prefix-tlv
             /ospfv3-e-lsa:sub-tlvs:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
          +--ro ospfv3-prefix-sid-flags
          |  +--ro flag*   identityref
          +--ro algorithm?                 identityref
          +--ro (sid)?
             +--:(sid-label)
             |  +--ro label-value?         uint32
             +--:(sid-index)
                +--ro index-value?         uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body
             /ospfv3-e-lsa:e-intra-area-prefix
             /ospfv3-e-lsa:e-intra-prefix-tlvs
             /ospfv3-e-lsa:intra-prefix-tlv/ospfv3-e-lsa:sub-tlvs:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
          +--ro ospfv3-prefix-sid-flags
          |  +--ro flag*   identityref
          +--ro algorithm?                 identityref
          +--ro (sid)?
             +--:(sid-label)
             |  +--ro label-value?         uint32
             +--:(sid-index)
                +--ro index-value?         uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body
```

```
             /ospfv3-e-lsa:e-inter-area-prefix
             /ospfv3-e-lsa:e-inter-prefix-tlvs
             /ospfv3-e-lsa:inter-prefix-tlv/ospfv3-e-lsa:sub-tlvs:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
           +--ro ospfv3-prefix-sid-flags
           |  +--ro flag*   identityref
           +--ro algorithm?                 identityref
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?         uint32
              +--:(sid-index)
                 +--ro index-value?         uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:database
             /ospf:as-scope-lsa-type/ospf:as-scope-lsas
             /ospf:as-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
             /ospf:body/ospfv3-e-lsa:e-as-external
             /ospfv3-e-lsa:e-external-tlvs
             /ospfv3-e-lsa:external-prefix-tlv/ospfv3-e-lsa:sub-tlvs:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
           +--ro ospfv3-prefix-sid-flags
           |  +--ro flag*   identityref
           +--ro algorithm?                 identityref
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?         uint32
              +--:(sid-index)
                 +--ro index-value?         uint32
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-nssa
             /ospfv3-e-lsa:e-external-tlvs
             /ospfv3-e-lsa:external-prefix-tlv/ospfv3-e-lsa:sub-tlvs:
     +--ro prefix-sid-sub-tlvs
        +--ro prefix-sid-sub-tlv* []
           +--ro ospfv3-prefix-sid-flags
           |  +--ro flag*   identityref
           +--ro algorithm?                 identityref
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?         uint32
              +--:(sid-index)
                 +--ro index-value?         uint32
   augment /rt:routing/rt:control-plane-protocols
```

```
             /rt:control-plane-protocol/ospf:ospf/ospf:areas
             /ospf:area/ospf:database/ospf:area-scope-lsa-type
             /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
             /ospf:ospfv3/ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router
             /ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv
             /ospfv3-e-lsa:sub-tlvs:
     +--ro adj-sid-sub-tlvs
     |  +--ro adj-sid-sub-tlv* []
     |     +--ro adj-sid-flags
     |     |  +--ro flag*   identityref
     |     +--ro weight?              uint8
     |     +--ro (sid)?
     |        +--:(sid-label)
     |        |  +--ro label-value?   uint32
     |        +--:(sid-index)
     |           +--ro index-value?   uint32
     +--ro lan-adj-sid-sub-tlvs
        +--ro lan-adj-sid-sub-tlv* []
           +--ro lan-adj-sid-flags
           |  +--ro flag*   identityref
           +--ro weight?               uint8
           +--ro neighbor-router-id?   rt-types:router-id
           +--ro (sid)?
              +--:(sid-label)
              |  +--ro label-value?    uint32
              +--:(sid-index)
                 +--ro index-value?    uint32
```

Authors' Addresses

Yingzhen Qu
Futurewei Technologies
Email: yingzhen.ietf@gmail.com

Acee Lindem
LabN Consulting, L.L.C.
301 Midenhall Way
Cary, NC 27513
Email: acee.ietf@gmail.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America
Email: zzhang@juniper.net

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

Internet
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2020

D. Yeung
Arrcus
Y. Qu
Futurewei
J. Zhang
Juniper Networks
I. Chen
The MITRE Corporation
A. Lindem
Cisco Systems
October 17, 2019

YANG Data Model for OSPF Protocol
draft-ietf-ospf-yang-29

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. 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
Appendix 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-protcol 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].

3. explicit-router-id: Support explicit per-instance Router-ID specification.

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].

26. ietf-spf-delay: Support IETF SPF delay algorithm [RFC8405].

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
          |     +--ro next-hops
          |     |  +--ro next-hop* [next-hop]
          |     |     +--ro outgoing-interface?   if:interface-ref
          |     |     +--ro next-hop              inet:ip-address
          |     +--ro metric?       uint32
          |     +--ro route-type?   route-type
          |     +--ro route-tag?    uint32
          +--ro statistics
          |  +--ro 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-chksum-sum?   uint32
          |  +--ro database
          |     +--ro as-scope-lsa-type*
          |        +--ro lsa-type?        uint16
          |        +--ro lsa-count?       yang:gauge32
          |        +--ro lsa-cksum-sum?   int32
          +--ro database
          |  +--ro as-scope-lsa-type* [lsa-type]
          |     +--ro as-scope-lsas
          |        +--ro as-scope-lsa* [lsa-id adv-router]
          |           +--ro lsa-id               union
          |           +--ro adv-router           inet:ipv4-address
          |           +--ro decoded-completed?   boolean
          |           +--ro raw-data?            yang:hex-string
          |           +--ro (version)?
          |              +--:(ospfv2)
          |              |  +--ro ospfv2
          .              .
          .              .
          |              +--:(ospfv3)
          |                 +--ro ospfv3
```

```
          .
          .
          +--ro spf-log
          |  +--ro event* [id]
          |     +--ro id                    uint32
          |     +--ro spf-type?             enumeration
          |     +--ro schedule-timestamp?   yang:timestamp
          |     +--ro start-timestamp?      yang:timestamp
                +--ro end-timestamp?        yang:timestamp
          |     +--ro trigger-lsa*
          |        +--ro area-id?      area-id-type
          |        +--ro link-id?      union
          |        +--ro type?         uint16
          |        +--ro lsa-id?       yang:dotted-quad
          |        +--ro adv-router?   yang:dotted-quad
          |        +--ro seq-num?      uint32
          +--ro lsa-log
          |  +--ro event* [id]
          |     +--ro id                    uint32
          |     +--ro lsa
          |     |  +--ro area-id?      area-id-type
          |     |  +--ro link-id?      union
          |     |  +--ro type?         uint16
          |     |  +--ro lsa-id?       yang:dotted-quad
          |     |  +--ro adv-router?   yang:dotted-quad
          |     |  +--ro seq-num?      uint32
          |     +--ro received-timestamp?   yang:timestamp
          |     +--ro reason?               identityref
          .
          .
```

2.6. 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).

```
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  -> ../../../../../../../../
          |      |                    ../../../rt:ribs/rib/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?
          |      |                                     yang:counter32
          |      |  +--ro area-scope-lsa-count?        yang:gauge32
          |      |  +--ro area-scope-lsa-cksum-sum?    int32
          |      |  +--ro database
          |      |     +--ro area-scope-lsa-type*
          |      |        +--ro lsa-type?        uint16
          |      |        +--ro lsa-count?       yang:gauge32
          |      |        +--ro lsa-cksum-sum?   int32
          |      +--ro database
          |      |  +--ro area-scope-lsa-type* [lsa-type]
          |      |     +--ro lsa-type           uint16
          |      |     +--ro area-scope-lsas
          |      |        +--ro area-scope-lsa* [lsa-id adv-router]
          |      |           +--ro lsa-id               union
        .      .           .
        .      .           .
          |      |           +--ro (version)?
          |      |              +--:(ospfv2)
          |      |              |  +--ro ospfv2
          |      |              |     +--ro header
        .      .              .     .
        .      .              .     .
          |      |              |     +--ro body
          |      |              |        +--ro router
```

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

```
|       |  +--rw virtual-link* [transit-area-id router-id]
|       |     +--rw transit-area-id       -> ../../../../
|       |                                    area/area-id
|       |     +--rw router-id             rt-types:router-id
|       |     +--rw hello-interval?       uint16
|       |     +--rw dead-interval?        uint32
|       |     +--rw retransmit-interval?  uint16
|       |     +--rw transmit-delay?       uint16
|       |     +--rw lls?                  boolean {lls}?
|       |     +--rw ttl-security {ttl-security}?
|       |     |  +--rw enable?   boolean
|       |     |  +--rw hops?     uint8
|       |     +--rw enable?               boolean
|       |     +--rw authentication
|       |     |  +--rw (auth-type-selection)?
|       |     |     +--:(ospfv2-auth)
|       |     |     |  +--rw ospfv2-auth-trailer-rfc?
|       |     |     |  |       ospfv2-auth-trailer-rfc-version
|       |     |     |  |        {ospfv2-authentication-trailer}?
|       |     |     |  +--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)
|       |     |     |      {ospfv3-authentication-ipsec}?
|       |     |     |  +--rw sa?                      string
|       |     |     +--:(ospfv3-auth-trailer)
|       |     |        |  {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
          |      |      |                        rt-types:router-id
          |      |      |     +--ro address?        inet:ip-address
          |      |      |     +--ro dr-router-id?   rt-types:router-id
          |      |      |     +--ro dr-ip-addr?     inet:ip-address
          |      |      |     +--ro bdr-router-id?  rt-types:router-id
          |      |      |     +--ro bdr-ip-addr?    inet:ip-address
          |      |      |     +--ro state?          nbr-state-type
          |      |      |     +--ro dead-timer? 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
```

```
|      |      +--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}?
|      |      |      |  +--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)
|      |      |      |       {ospfv3-authentication-ipsec}?
|      |      |      |  +--rw sa?                       string
|      |      |      +--:(ospfv3-auth-trailer)
|      |      |         |  {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
|      |      +--rw cost?               uint16
|      |      +--rw mtu-ignore?         boolean
|      |                                {mtu-ignore}?
|      |      +--rw prefix-suppression? boolean
|      |                                {prefix-suppression}?
|      |      +--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
          |      |      |                          rt-types:router-id
          |      |      |     +--ro address?        inet:ip-address
          |      |      |     +--ro dr-router-id?   rt-types:router-id
          |      |      |     +--ro dr-ip-addr?     inet:ip-address
          |      |      |     +--ro bdr-router-id?  rt-types:router-id
          |      |      |     +--ro bdr-ip-addr?    inet:ip-address
          |      |      |     +--ro state?          nbr-state-type
          |      |      |     +--ro cost?           uint32
          |      |      |     +--ro dead-timer? rt-types:
          |      |      |     |                  rtimer-value-seconds16
          |      |      |     +--ro statistics
          |      |      |        +--ro nbr-event-count?
          |      |      |                          yang:counter32
          |      |      |        +--ro nbr-retrans-qlen?
          |      |      |                          yang:gauge32
          |      |      +--ro database
          |      |         +--ro link-scope-lsa-type* [lsa-type]
          |      |            +--ro lsa-type           uint16
          |      |            +--ro link-scope-lsas
          |      |
        .      .
        .      .
```

2.7. 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}?
```

```
|            |      |  +--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)
|            |      |      {ospfv3-authentication-ipsec}?
|            |      |  +--rw sa?                       string
|            |      +--:(ospfv3-auth-trailer)
|            |         |  {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
|            +--rw cost?               uint16
|            +--rw mtu-ignore?         boolean
|            |                         {mtu-ignore}?
|            +--rw prefix-suppression? boolean
|            |                         {prefix-suppression}?
|            +--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 if-event-count?        yang:counter32
|            |  +--ro link-scope-lsa-count?  yang:gauge32
|            |  +--ro link-scope-lsa-cksum-sum?
|            |                               uint32
|            |  +--ro database
|            |     +--ro link-scope-lsa-type*
|            |        +--ro lsa-type?        uint16
|            |        +--ro lsa-count?       yang:gauge32
|            |        +--ro lsa-cksum-sum?   int32
|            +--ro neighbors
```

```
          |            |  +--ro neighbor* [neighbor-router-id]
          |            |     +--ro neighbor-router-id
          |            |                          rt-types:router-id
          |            |     +--ro address?        inet:ip-address
          |            |     +--ro dr-router-id?   rt-types:router-id
          |            |     +--ro dr-ip-addr?     inet:ip-address
          |            |     +--ro bdr-router-id?  rt-types:router-id
          |            |     +--ro bdr-ip-addr?    inet:ip-address
          |            |     +--ro state?          nbr-state-type
          |            |     +--ro dead-timer? rt-types:
          |            |     |                  rtimer-value-seconds16
          |            |     +--ro statistics
          |            |        +--ro nbr-event-count?
          |            |                          yang:counter32
          |            |        +--ro nbr-retrans-qlen?
          |            |                          yang:gauge32
          |            +--ro database
          |            .  +--ro link-scope-lsa-type* [lsa-type]
          |            .     +--ro lsa-type           uint16
          |            .     +--ro link-scope-lsas
        .            .
        .            .
          |            +--rw topologies {ospf:multi-topology}?
          |            |  +--rw topology* [name]
          |            |     +--rw name  -> ../../../../../../../../
          |            |                    ../../../rt:ribs/rib/name
          |            |     +--rw cost? uint32
          |            +--rw instance-id?           uint8
        .
        .
```

2.8. 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]/
    |  +         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 state?                   if-state-type
       +---n if-config-error
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/af
       |  +--ro (if-link-type-selection)?
       |  |  +--:(interface)
       |  |  |  +--ro interface
       |  |  |     +--ro interface?   if:interface-ref
       |  |  +--:(virtual-link)
       |  |  |  +--ro virtual-link
       |  |  |     +--ro transit-area-id?      area-id-type
       |  |  |     +--ro neighbor-router-id?   rt-types:router-id
       |  |  +--:(sham-link)
       |  |     +--ro sham-link
       |  |        +--ro area-id?          area-id-type
       |  |        +--ro local-ip-addr?    inet:ip-address
       |  |        +--ro remote-ip-addr?   inet:ip-address
       |  +--ro packet-source?           yang:dotted-quad
       |  +--ro packet-type?             packet-type
       |  +--ro error?                   enumeration
       +---n nbr-state-change
       |  +--ro routing-protocol-name?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol/name
       |  +--ro af?
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/af
```

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

```
       |  +     -> /rt:routing/control-plane-protocols/
       |  +         control-plane-protocol
       |  +         [rt:name=current()/../routing-protocol-name]/
       |  +         ospf:ospf/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
```

```
       |  +--ro area-id?                 area-id-type
       |  +--ro status?                  nssa-translator-state-type
       +---n restart-status-change
          +--ro routing-protocol-name?
          +     -> /rt:routing/control-plane-protocols/
          +         control-plane-protocol/name
          +--ro af?
          +     -> /rt:routing/control-plane-protocols/
          +         control-plane-protocol
          +         [rt:name=current()/../routing-protocol-name]/
          +         ospf:ospf/af
          +--ro status?                   restart-status-type
          +--ro restart-interval?         uint16
          +--ro exit-reason?              restart-exit-reason-type
```

2.9. OSPF RPC Operations

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

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

- o clear-neighbor: Reset a particular OSPF neighbor or group of neighbors associated with an OSPF interface.

```
     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], [RFC7770], [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
        (https://trustee.ietf.org/license-info).

        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.

        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 (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
```

```
          will accept.";
        reference "RFC 1765: OSPF Database Overload";
      }

      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
                   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
                   RFC 7474: Security Extension for OSPFv2 When
                   Using Manual Key Management";
      }

      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
```

```
        "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, wnly 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.";
```

```
        reference "RFC 905: ISO Transport Protocol specification
                   ISO DP 8073";
     }

     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";
           reference
             "RFC 7474: Security Extension for OSPFv2
              When Using Manual Key Management Authentication";

         }
       }
       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.";
        }
        description "Inter-Area-Router 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 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 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;
        }
      }
    }

    grouping lsa-key {
      description
        "OSPF LSA key - the database key for each LSA of a given
         type in the Link State DataBase (LSDB).";
      leaf lsa-id {
        type union {
          type yang:dotted-quad;
          type uint32;
        }
        description
          "Link-State ID.";
      }
      leaf adv-router {
        type rt-types:router-id;
        description
          "Advertising router.";
      }
    }

    grouping instance-stat {
      description "Per-instance 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 instance's counters suffered a
           discontinuity.  If no such discontinuities have occurred
           since the OSPF instance was last re-initialized, then
           this node contains the time the OSPF instance was
           re-initialized which normally occurs when it was
           created.";
      }
      leaf originate-new-lsa-count {
        type yang:counter32;
        description
          "The number of new LSAs originated. Discontinuities in the
           value of this counter can occur when the OSPF instance is
           re-initialized.";
      }
      leaf rx-new-lsas-count {
```

```
        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.";
      }
      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 andcreated.";
      }
      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
```

```
                 (like RSVP-TE or GRE).";
            }
            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.";
      }
      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 lls {
         if-feature lls;
         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 OSFPv2 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.";
            }
          }
          description
            "Interface type.";
        }

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

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

        leaf priority {
          type uint8;
          description
            "Configure OSPF router priority. 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')" {
```

```
                    description "OSPFv3 LSA.";
                  }
                }
              }
            }
          }
        }
      }
    }

    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 {
```

```
      description
        "OSPF sham link operational state.";
      uses interface-common-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.";
          }
        }
      }
    }

    grouping area-common-config {
      description
        "OSPF area common configuration state.";

      leaf summary {
        when "derived-from(../area-type,'stub-nssa-area')" {
          description
            "Summary advertisement into the stub/NSSA area.";
        }
        type boolean;
        description
          "Enable/Disable summary advertisement into the stub or
```

```
            NSSA area.";
      }
      leaf default-cost {
        when "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.";
          }
          container next-hops {
```

```
            description "Next hops for the route.";
            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
```

```
          "Delay used while in LONG_WAIT state (milliseconds).";
      }
      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
```

```
          "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 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
```

```
             Engineering (TE)";
          leaf ipv4-router-id {
            type inet:ipv4-address;
            description
              "Explicitly configure the TE IPv4 Router ID.";
          }
          leaf ipv6-router-id {
            type inet:ipv6-address;
            description
              "Explicitly configure the TE IPv6 Router ID.";
          }
        }
        container ldp {
          description
            "OSPF MPLS LDP config state.";
          leaf igp-sync {
            if-feature ldp-igp-sync;
            type boolean;
            description
              "Enable LDP IGP synchronization.";
          }
        }
      }
      uses instance-fast-reroute-config;
      uses node-tag-config;
    }

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

      leaf router-id {
        type rt-types:router-id;
        config false;
        description
          "Defined in RFC 2328. A 32-bit number
           that uniquely identifies the router.";
      }

      uses local-rib;

      container statistics {
        config false;
        description "Per-instance statistics";
        uses instance-stat;
      }

      container 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 "All AS-scope of LSA 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;
```

```
        description
          "Enable/Disable summary advertisement into the
           topology in the stub or NSSA area.";
      }
      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
```

```
          "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;
```

```
            description
              "This is the timestamp when the computation was
               scheduled.";
          }
          leaf start-timestamp {
            type yang:timestamp;
            description
              "This is the timestamp when the computation was
               started.";
          }
          leaf end-timestamp {
            type yang:timestamp;
            description
              "This 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.";
```

```
            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[area-id=current()]/area-type, "
                    + "'normal-area') and "
                    + "../../../../area[area-id=current()]/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-seconds16;
          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:

/ospf/database

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

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

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

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

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 reconstucted. 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:

```
URI: urn:ietf:params:xml:ns:yang:ietf-ospf
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
namespace: urn:ietf:params:xml:ns:yang: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]
Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work in progress), August 2018.

[RFC1765] Moy, J., "OSPF Database Overflow", RFC 1765, DOI 10.17487/RFC1765, March 1995, <https://www.rfc-editor.org/info/rfc1765>.

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

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

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

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

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

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <https://www.rfc-editor.org/info/rfc4552>.

[RFC4576] Rosen, E., Psenak, P., and P. Pillay-Esnault, "Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4576, DOI 10.17487/RFC4576, June 2006, <https://www.rfc-editor.org/info/rfc4576>.

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

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

[RFC4973] Srisuresh, P. and P. Joseph, "OSPF-xTE: Experimental Extension to OSPF for Traffic Engineering", RFC 4973, DOI 10.17487/RFC4973, July 2007, <https://www.rfc-editor.org/info/rfc4973>.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <https://www.rfc-editor.org/info/rfc5082>.

[RFC5185] Mirtorabi, S., Psenak, P., Lindem, A., Ed., and A. Oswal, "OSPF Multi-Area Adjacency", RFC 5185, DOI 10.17487/RFC5185, May 2008, <https://www.rfc-editor.org/info/rfc5185>.

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

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

[RFC5309] Shen, N., Ed. and A. Zinin, Ed., "Point-to-Point Operation over LAN in Link State Routing Protocols", RFC 5309, DOI 10.17487/RFC5309, October 2008, <https://www.rfc-editor.org/info/rfc5309>.

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

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

[RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <https://www.rfc-editor.org/info/rfc5613>.

[RFC5642] Venkata, S., Harwani, S., Pignataro, C., and D. McPherson, "Dynamic Hostname Exchange Mechanism for OSPF", RFC 5642, DOI 10.17487/RFC5642, August 2009, <https://www.rfc-editor.org/info/rfc5642>.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, DOI 10.17487/RFC5709, October 2009, <https://www.rfc-editor.org/info/rfc5709>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <https://www.rfc-editor.org/info/rfc5714>.

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

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

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

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

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

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

[RFC6860] Yang, Y., Retana, A., and A. Roy, "Hiding Transit-Only Networks in OSPF", RFC 6860, DOI 10.17487/RFC6860, January 2013, <https://www.rfc-editor.org/info/rfc6860>.

[RFC6987] Retana, A., Nguyen, L., Zinin, A., White, R., and D. McPherson, "OSPF Stub Router Advertisement", RFC 6987, DOI 10.17487/RFC6987, September 2013, <https://www.rfc-editor.org/info/rfc6987>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <https://www.rfc-editor.org/info/rfc7166>.

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

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <https://www.rfc-editor.org/info/rfc7490>.

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

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

[RFC7777] Hegde, S., Shakir, R., Smirnov, A., Li, Z., and B. Decraene, "Advertising Node Administrative Tags in OSPF", RFC 7777, DOI 10.17487/RFC7777, March 2016, <https://www.rfc-editor.org/info/rfc7777>.

[RFC7884] Pignataro, C., Bhatia, M., Aldrin, S., and T. Ranganath, "OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators", RFC 7884, DOI 10.17487/RFC7884, July 2016, <https://www.rfc-editor.org/info/rfc7884>.

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

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8405] Decraene, B., Litkowski, S., Gredler, H., Lindem, A., Francois, P., and C. Bowers, "Shortest Path First (SPF) Back-Off Delay Algorithm for Link-State IGPs", RFC 8405, DOI 10.17487/RFC8405, June 2018, <https://www.rfc-editor.org/info/rfc8405>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <https://www.rfc-editor.org/info/rfc8476>.

7.2. Informative References

[RFC0905] "ISO Transport Protocol specification ISO DP 8073", RFC 905, DOI 10.17487/RFC0905, April 1984, <https://www.rfc-editor.org/info/rfc905>.

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

[RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, DOI 10.17487/RFC5443, March 2009, <https://www.rfc-editor.org/info/rfc5443>.

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

[RFC5649] Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, DOI 10.17487/RFC5649, September 2009, <https://www.rfc-editor.org/info/rfc5649>.

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

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

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

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

EMail: acee@cisco.com

Open Shortest Path First IGP
Internet-Draft
Intended status: Standards Track
Expires: November 24, 2017

P. Psenak, Ed.
K. Talaulikar
Cisco Systems, Inc.
W. Henderickx
Nokia
P. Pillay-Esnault
Huawei

May 23, 2017

OSPF LLS Extensions for Local Interface ID Advertisement
draft-ppsenak-ospf-lls-interface-id-01

Abstract

This draft describes the extensions to OSPF link-local signaling to advertise Local Interface Identifier.

Requirements Language

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

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 November 24, 2017.

Copyright Notice

Copyright (c) 2017 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 2
2. Interface ID Exchange using TE Opaque LSA 2
3. Interface ID Exchange using OSPF LLS 3
3.1. Local Interface Identifier TLV 3
4. Backward Compatibility with RFC 4203 4
5. IANA Considerations 4
6. Security Considerations 4
7. Contributors 4
8. Acknowledgements 4
9. Normative References 4
Authors' Addresses 5

1. Introduction

Every interface is assigned an Interface ID, which uniquely identifies the interface on the router. For example, some implementations MAY be able to use the MIB-II IfIndex [RFC2863] as the Interface ID.

Local/Remote Interface Identifiers MAY be flooded by OSPF [RFC2328] as defined in [RFC4203]. From the perspective of the advertising router, the Local Interface Identifier is a known value, however the Remote Interface Identifier needs to be learnt before it can be advertised. [RFC4203] suggests to use TE Link Local LSA [RFC3630] to communicate Local Interface Identifier to neighbors on the link. Though such mechanism works, it has some drawbacks.

This draft proposes an extension to OSPF link-local signaling (LLS) [RFC5613] to advertise the Local Interface Identifier.

2. Interface ID Exchange using TE Opaque LSA

Usage of the Link Local TE Opaque LSA to propagate the Local Interface Identifier to the neighbors on the link is described in [RFC4203]. This mechanism has following problems:

LSAs can only be flooded over an existing adjacency that is in Exchange state or greater. The adjacency state machine progresses

independently on each side of the adjacency and, as such, may reach the Full state on one side before the TE Link Opaque LSA arrives. The consequence is that link can be initially advertised without the Remote Interface Identifier. Later when the TE Link Opaque LSA arrives, the link must be advertised again, this time with the valid Remote Interface Identifier. Implementation may choose to wait before advertising the link, but there is no guarantee that the neighbor will ever advertise the TE Link Opaque LSA with the Interface Identifier. In summary, the existing mechanism does not guarantee that Remote Interface Identifier is known at the time the link is advertised.

TE Opaque LSA is defined for MPLS Traffic Engineering, but the knowledge of the Remote Interface Identifier is useful for other cases where MPLS TE is not used. One example is the lack of valid 2-way connectivity check for remote parallel point-to-point links in OSPF. In such case, TE Opaque LSAs are not exchanged solely for 2-way connectivity correctness.

3. Interface ID Exchange using OSPF LLS

To address the problems described earlier and to allow the Interface Identifiers exchange to be part of the neighbor discovery process, we propose to extend OSPF link-local signaling to advertise the Local Interface Identifier in OSPF Hello packets.

3.1. Local Interface Identifier TLV

The Local Interface Identifier TLV is a new LLS TLV. It has following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Local Interface Identifier                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: TBD, suggested value 18

Length: 4 octet

Local Interface Identifier: The value of the local Interface Identifier.

Local Interface Identifier TLV MUST be present in all Hello packets on all link types, except packets that are sent to the remote end of the virtual-link.

4. Backward Compatibility with RFC 4203

Implementations which support Local Interface ID signalling using LLS MUST prefer the Local Interface ID value received through LLS over the value received through the Link Local TE Opaque LSAs.

Implementations which also support the Local Interface ID signalling via Link Local TE Opaque LSA MAY continue to do so to ensure backward compatibility and they MUST signal the same local interface id via both mechanisms.

During the rare conditions, when the Local Interface ID changes, a timing interval may exist, where the received values of the Local Interface ID advertised through LLS and Link Local TE Opaque LSA may differ. Such situation is temporary and received values via both mechanisms should become equal as soon as the next Hello and/or Link Local TE Opaque LSA is re-generated by the originator.

5. IANA Considerations

This specification updates Link Local Signalling TLV Identifiers registry.

Following values is allocated:

o 18 - Local Interface Identifier TLV

6. Security Considerations

Implementations must assure that malformed LLS TLV and Sub-TLV permutations do not result in errors which cause hard OSPF failures.

7. Contributors

8. Acknowledgements

Thanks to Tony Przygienda for his extensive review and useful comments.

9. Normative References

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

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

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <http://www.rfc-editor.org/info/rfc2863>.

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

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <http://www.rfc-editor.org/info/rfc4203>.

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

[RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <http://www.rfc-editor.org/info/rfc5613>.

Authors' Addresses

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

Email: ppsenak@cisco.com

Ketan Jivan Talaulikar
Cisco Systems, Inc.
S.No. 154/6, Phase I, Hinjawadi
PUNE, MAHARASHTRA 411 057
India

Email: ketant@cisco.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
BE

Email: wim.henderickx@nokia.com

Padma Pillay-Esnault
Huawei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: padma@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 25, 2017

P. Psenak
A. Lindem
L. Ginsberg
Cisco Systems
W. Henderickx
Nokia
J. Tantsura
Individual
H. Gredler
RtBrick Inc.
June 23, 2017

OSPFv2 Link Traffic Engineering (TE) Attribute Reuse
draft-ppsenak-ospf-te-link-attr-reuse-05.txt

Abstract

Various link attributes have been defined in OSPFv2 in the context of the MPLS Traffic Engineering (TE) and GMPLS. Many of these link attributes can be used for purposes other than MPLS Traffic Engineering or GMPLS. This documents defines how to distribute such attributes in OSPFv2 for applications other than MPLS Traffic Engineering or GMPLS purposes.

Status of This Memo

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 December 25, 2017.

Copyright Notice

Copyright (c) 2017 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 3
  1.1. Requirements notation 3
2. Link attributes examples 3
3. Advertising Link Attributes 4
  3.1. TE Opaque LSA 4
  3.2. Extended Link Opaque LSA 5
  3.3. Selected Approach 5
4. Reused TE link attributes 6
  4.1. Remote interface IP address 6
  4.2. Link Local/Remote Identifiers 6
  4.3. Shared Risk Link Group (SRLG) 7
  4.4. Extended Metrics 7
5. Advertisement of Application Specific Values 7
6. Deployment Considerations 10
7. Attribute Advertisements and Enablement 10
8. Backward Compatibility 11
9. Security Considerations 11
10. IANA Considerations 12
11. Acknowledgments 12
12. References 12
  12.1. Normative References 12
  12.2. Informative References 13
Authors' Addresses 14

1. Introduction

Various link attributes have been defined in OSPFv2 [RFC2328] in the context of the MPLS traffic engineering and GMPLS. All these attributes are distributed by OSPFv2 as sub-TLVs of the Link-TLV advertised in the OSPFv2 TE Opaque LSA [RFC3630].

Many of these link attributes are useful outside of the traditional MPLS Traffic Engineering or GMPLS. This brings its own set of problems, in particular how to distribute these link attributes in OSPFv2 when MPLS TE or GMPLS are not deployed or are deployed in parallel with other applications that use these link attributes.

[RFC7855] discusses use cases/requirements for SR. Included among these use cases is SRTE. If both RSVP-TE and SRTE are deployed in a network, link attribute advertisements can be used by one or both of these applications. As there is no requirement for the link attributes advertised on a given link used by SRTE to be identical to the link attributes advertised on that same link used by RSVP-TE, there is a clear requirement to indicate independently which link attribute advertisements are to be used by each application.

As the number of applications which may wish to utilize link attributes may grow in the future, an additional requirement is that the extensions defined allow the association of additional applications to link attributes without altering the format of the advertisements or introducing new backwards compatibility issues.

Finally, there may still be many cases where a single attribute value can be shared among multiple applications, so the solution should minimize advertising duplicate link/attribute when possible.

1.1. Requirements notation

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

2. Link attributes examples

This section lists some of the link attributes originally defined for MPLS Traffic Engineering that can be used for other purposes in OSPFv2. The list doesn’t necessarily contain all the required attributes.

1. Remote Interface IP address [RFC3630] - OSPFv2 currently cannot distinguish between parallel links between two OSPFv2 routers. As a result, the two-way connectivity check performed during SPF

may succeed when the two routers disagree on which of the links to use for data traffic.

2. Link Local/Remote Identifiers - [RFC4203] - Used for the two-way connectivity check for parallel unnumbered links. Also used for identifying adjacencies for unnumbered links in Segment Routing traffic engineering.

3. Shared Risk Link Group (SRLG) [RFC4203] - In IPFRR, the SRLG is used to compute diverse backup paths [RFC5714].

4. Unidirectional Link Delay/Loss Metrics [RFC7471] - Could be used for the shortest path first (SPF) computation using alternate metrics within an OSPF area.

3. Advertising Link Attributes

This section outlines possible approaches for advertising link attributes originally defined for MPLS Traffic Engineering purposes or GMPLS when they are used for other applications.

3.1. TE Opaque LSA

One approach for advertising link attributes is to continue to use TE Opaque LSA ([RFC3630]). There are several problems with this approach:

1. Whenever the link is advertised in a TE Opaque LSA, the link becomes a part of the TE topology, which may not match IP routed topology. By making the link part of the TE topology, remote nodes may mistakenly believe that the link is available for MPLS TE or GMPLS, when, in fact, MPLS is not enabled on the link.

2. The TE Opaque LSA carries link attributes that are not used or required by MPLS TE or GMPLS. There is no mechanism in a TE Opaque LSA to indicate which of the link attributes are passed to MPLS TE application and which are used by other applications including OSPFv2 itself.

3. Link attributes used for non-TE purposes are partitioned across multiple LSAs - the TE Opaque LSA and the Extended Link Opaque LSA. This partitioning will require implementations to lookup multiple LSAs to extract link attributes for a single link, bringing needless complexity to OSPFv2 implementations.

The advantage of this approach is that there is no additional standardization requirement to advertise the TE/GMPL attributes for other applications. Additionally, link attributes are only

advertised once when both OSPF TE and other applications are deployed on the same link. This is not expected to be a common deployment scenario.

3.2. Extended Link Opaque LSA

An alternative approach for advertising link attributes is to use Extended Link Opaque LSAs as defined in [RFC7684]. This LSA was defined as a generic container for distribution of the extended link attributes. There are several advantages in using Extended Link LSA:

1. Advertisement of the link attributes does not make the link part of the TE topology. It avoids any conflicts and is fully compatible with the [RFC3630].

2. The TE Opaque LSA remains truly opaque to OSPFv2 as originally defined in [RFC3630]. Its content is not inspected by OSPFv2 and OSPFv2 acts as a pure transport.

3. There is clear distinction between link attributes used by TE and link attributes used by other OSPFv2 applications.

4. All link attributes that are used by OSPFv2 applications are advertised in a single LSA, the Extended Link Opaque LSA.

The disadvantage of this approach is that in rare cases, the same link attribute is advertised in both the TE Opaque and Extended Link Attribute LSAs. Additionally, there will be additional standardization effort. However, this could also be viewed as an advantage as the non-TE use cases for the TE link attributes are documented and validated by the OSPF working group.

3.3. Selected Approach

It is RECOMMENDED to use the Extended Link Opaque LSA ([RFC7684] to advertise any link attributes used for non-TE purposes in OSPFv2, including those that have been originally defined for TE purposes. TE link attributes used for TE purposes continue to use TE Opaque LSA ([RFC3630]).

It is also RECOMMENDED to keep the format of the link attribute TLVs that have been defined for TE purposes unchanged even when they are used for non-TE purposes.

Finally, it is RECOMMENDED to allocate unique code points for link attribute TLVs that have been defined for TE purposes for the OSPFv2 Extended Link TLV Sub-TLV Registry as defined in [RFC7684]. For each

reused TLV, the code point will be defined in an IETF document along with the expected usecase(s).

4. Reused TE link attributes

This section defines the use case and code points for the OSPFv2 Extended Link TLV Sub-TLV Registry for some of the link attributes that have been originally defined for TE or GMPLS purposes.

4.1. Remote interface IP address

The OSPFv2 description of an IP numbered point-to-point adjacency does not include the remote IP address. As described in Section 2, this makes the two-way connectivity check ambiguous in the presence of the parallel point-to-point links between two OSPFv2 routers.

The Remote IP address of the link can also be used for Segment Routing traffic engineering to identify the link in a set of parallel links between two OSPFv2 routers [I-D.ietf-ospf-segment-routing-extensions]. Similarly, the remote IP address is useful in identifying individual parallel OSPF links advertised in BGP Link-State as described in [I-D.ietf-idr-ls-distribution].

To advertise the Remote interface IP address in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 2.5.4. of [RFC3630] is used and TLV type TBD1 is used.

4.2. Link Local/Remote Identifiers

The OSPFv2 description of an IP unnumbered point-to-point adjacency does not include the remote link identifier. As described in Section 2, this makes the two-way connectivity check ambiguous in the presence of the parallel point-to-point IP unnumbered links between two OSPFv2 routers.

The local and remote link identifiers can also be used for Segment Routing traffic engineering to identify the link in a set of parallel IP unnumbered links between two OSPFv2 routers [I-D.ietf-ospf-segment-routing-extensions]. Similarly, these identifiers are useful in identifying individual parallel OSPF links advertised in BGP Link-State as described in [I-D.ietf-idr-ls-distribution].

To advertise the link Local/Remote identifiers in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 1.1. of [RFC4203] is used and TLV type TBD2 is used.

4.3. Shared Risk Link Group (SRLG)

The SRLG of a link can be used in IPFRR to compute a backup path that does not share any SRLG group with the protected link.

To advertise the SRLG of the link in the OSPFv2 Extended Link TLV, the same format of the sub-TLV as defined in section 1.3. of [RFC4203] is used and TLV type TBD3 is used.

4.4. Extended Metrics

[RFC3630] defines several link bandwidth types. [RFC7471] defines extended link metrics that are based on link bandwidth, delay and loss characteristics. All these can be used to compute best paths within an OSPF area to satisfy requirements for bandwidth, delay (nominal or worst case) or loss.

To advertise extended link metrics in the OSPFv2 Extended Link TLV, the same format of the sub-TLVs as defined in [RFC7471] is used with following TLV types:

TBD4 - Unidirectional Link Delay

TBD5 - Min/Max Unidirectional Link Delay

TBD6 - Unidirectional Delay Variation

TBD7 - Unidirectional Link Loss

TBD8 - Unidirectional Residual Bandwidth

TBD9 - Unidirectional Available Bandwidth

TBD10 - Unidirectional Utilized Bandwidth

5. Advertisement of Application Specific Values

Multiple applications can utilize link attributes that are flooded by OSPFv2. Some examples of applications using the link attributes are Segment Routing Traffic Engineering and LFA [RFC5286].

In some cases the link attribute only has a single value that is applicable to all applications. An example is a Remote interface IP address [Section 4.1] or Link Local/Remote Identifiers [Section 4.2].

In some cases the link attribute MAY have different values for different applications. An example could be SRLG [Section 4.3],

where values used by LFA could be different then the values used by Segment Routing Traffic Engineering.

To allow advertisement of the application specific values of the link attribute, a new Extended Link Attribute sub-TLV of the Extended Link TLV [RFC7471] is defined. The Extended Link Attribute sub-TLV is an optional sub-TLV and can appear multiple times in the Extended Link TLV. It has following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     SABML     |     UDABML    |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Standard Application Bit-Mask                  |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                User Defined Application Bit-Mask              |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link Attribute sub-sub-TLVs              |
+-                                                             -+
|                            ...                                |
```

where:

Type: TBD11, suggested value 14

Length: variable

SABML: Standard Application Bit-Mask Length. If the Standard Application Bit-Mask is not present, the Standard Application Bit-Mask Length MUST be set to 0.

UDABML: User Defined Application Bit-Mask Length. If the User Defined Application Bit-Mask is not present, the User Defined Application Bit-Mask Length MUST be set to 0.

Standard Application Bit-Mask: Optional set of bits, where each bit represents a single standard application. The following bits are defined by this document:

Bit-0: RSVP Traffic Engineering

Bit-1: Segment Routing Traffic Engineering

Bit-2: Loop Free Alternate (LFA). Includes all LFA types.

User Defined Application Bit-Mask: Optional set of bits, where each bit represents a single user defined application.

Standard Application Bits are defined/sent starting with Bit 0. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of octets that will need to be transmitted.

User Defined Application bits have no relationship to Standard Application bits and are NOT managed by IANA or any other standards body. It is recommended that bits are used starting with Bit 0 so as to minimize the number of octets required to advertise all of them.

Undefined bits in both Bit-Masks MUST be transmitted as 0 and MUST be ignored on receipt. Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

If the link attribute advertisement is limited to be used by a specific set of applications, corresponding Bit-Masks MUST be present and application specific bit(s) MUST be set for all applications that use the link attributes advertised in the Extended Link Attribute sub-TLV.

Application Bit-Masks apply to all link attributes that support application specific values and are advertised in the Extended Link Attribute sub-TLV.

The advantage of not making the Application Bit-Masks part of the attribute advertisement itself is that we can keep the format of the link attributes that have been defined previously and reuse the same format when advertising them in the Extended Link Attribute sub-TLV.

If the link attribute is advertised and there is no Application Bit-Mask present in the Extended Link Attribute Sub-TLV, the link attribute advertisement MAY be used by any application. If, however, another advertisement of the same link attribute includes any Application Bit-Mask in the Extended Link Attribute sub-TLV, applications that are listed in the Application Bit-Masks of such Extended Link Attribute sub-TLV SHOULD use the attribute advertisement which has the application specific bit set in the Application Bit-Masks.

If the same application is listed in the Application Bit-Masks of more then one Extended Link Attribute sub-TLV, the application SHOULD

use the first advertisement and ignore any subsequent advertisements of the same attribute. This situation SHOULD be logged as an error.

This document defines the set of link attributes for which the Application Bit-Masks may be advertised. If any of the Application Bit-Masks is included in the Extended Link Attribute sub-TLV that advertises any link attribute(s) NOT listed below, the Application Bit-Masks MUST NOT be used for such link attribute(s). It MUST be used for those attribute(s) that support application specific values. Documents which define new link attributes MUST state whether the new attributes support application specific values. The link attributes to which the Application Bit-Masks may apply are:

- Shared Risk Link Group

- Unidirectional Link Delay

- Min/Max Unidirectional Link Delay

- Unidirectional Delay Variation

- Unidirectional Link Loss

- Unidirectional Residual Bandwidth

- Unidirectional Available Bandwidth

- Unidirectional Utilized Bandwidth

6. Deployment Considerations

If link attributes are advertised associated with zero length application bit masks for both standard applications and user defined applications, then that set of link attributes MAY be used by any application. If support for a new application is introduced on any node in a network in the presence of such advertisements, these advertisements MAY be used by the new application. If this is not what is intended, then existing advertisements MUST be readvertised with an explicit set of applications specified before a new application is introduced.

7. Attribute Advertisements and Enablement

This document defines extensions to support the advertisement of application specific link attributes. The presence or absence of link attribute advertisements for a given application on a link does NOT indicate the state of enablement of that application on that

link. Enablement of an application on a link is controlled by other means.

For some applications, the concept of enablement is implicit. For example, SRTE implicitly is enabled on all links which are part of the Segment Routing enabled topology. Advertisement of link attributes supports constraints which may be applied when specifying an explicit path through that topology.

For other applications enablement is controlled by local configuration. For example, use of a link as an LFA can be controlled by local enablement/disablement and/or the use of administrative tags.

It is an application specific policy as to whether a given link can be used by that application even in the absence of any application specific link attributes.

8. Backward Compatibility

Link attributes may be concurrently advertised in both the TE Opaque LSA [RFC3630] and the Extended Link Opaque LSA [RFC7684].

In fact, there is at least one OSPF implementation that utilizes the link attributes advertised in TE Opaque LSAs [RFC3630] for Non-RSVP TE applications. For example, this implementation of LFA and remote LFA utilizes links attributes such as Shared Risk Link Groups (SRLG) [RFC4203] and Admin Group [[RFC3630]advertised in TE Opaque LSAs. These applications are described in [RFC5286], [RFC7490], [I-D.ietf-rtgwg-lfa-manageability] and [I-D.psarkar-rtgwg-rlfa-node-protection].

When an OSPF routing domain includes routers using link attributes from TE Opaque LSAs for Non-RSVP TE applications such as LFA, OSPF routers in that domain should continue to advertise such TE Opaque LSAs. If there are also OSPF routers using the link attributes described herein for any application, OSPF routers in the routing domain will also need to advertise these attributes in OSPF Extended Link Attributes LSAs [RFC7684]. In such a deployment, the advertised attributes SHOULD be the same and Non-RSVP application access to link attributes is a matter of local policy.

9. Security Considerations

Implementations must assure that malformed TLV and Sub-TLV permutations do not result in errors that cause hard OSPFv2 failures.

10. IANA Considerations

OSPFv2 Extended Link TLV Sub-TLVs registry [RFC7684] defines sub-TLVs at any level of nesting for OSPFv2 Extended Link TLVs. This specification updates OSPFv2 Extended Link TLV sub-TLVs registry with the following TLV types:

TBD1 (4 Recommended) - Remote interface IP address

TBD2 (5 Recommended) - Link Local/Remote Identifiers

TBD3 (6 Recommended) - Shared Risk Link Group

TBD4 (7 Recommended) - Unidirectional Link Delay

TBD5 (8 Recommended) - Min/Max Unidirectional Link Delay

TBD6 (9 Recommended) - Unidirectional Delay Variation

TBD7 (10 Recommended) - Unidirectional Link Loss

TBD8 (11 Recommended) - Unidirectional Residual Bandwidth

TBD9 (12 Recommended) - Unidirectional Available Bandwidth

TBD10 (13 Recommended) - Unidirectional Utilized Bandwidth

TBD11 (14 Recommended) - Extended Link Attribute

This specification defines a new Link-Attribute-Applicability Application Bits registry and defines following bits:

Bit-0 - Segment Routing Traffic Engineering

Bit-1 - LFA

11. Acknowledgments

Thanks to Chris Bowers for his review and comments.

12. References

12.1. Normative References

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

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

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <http://www.rfc-editor.org/info/rfc5714>.

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

12.2. Informative References

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-13 (work in progress), October 2015.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-16 (work in progress), May 2017.

[I-D.ietf-rtgwg-lfa-manageability]
Litkowski, S., Decraene, B., Filsfils, C., Raza, K., and M. Horneffer, "Operational management of Loop Free Alternates", draft-ietf-rtgwg-lfa-manageability-11 (work in progress), June 2015.

[I-D.psarkar-rtgwg-rlfa-node-protection]
psarkar@juniper.net, p., Gredler, H., Hegde, S., Bowers, C., Litkowski, S., and H. Raghuveer, "Remote-LFA Node Protection and Manageability", draft-psarkar-rtgwg-rlfa-node-protection-05 (work in progress), June 2014.

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

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <http://www.rfc-editor.org/info/rfc4203>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <http://www.rfc-editor.org/info/rfc5286>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <http://www.rfc-editor.org/info/rfc7471>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <http://www.rfc-editor.org/info/rfc7490>.

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

Authors' Addresses

Peter Psenak
Cisco Systems
Apollo Business Center
Mlynske nivy 43
Bratislava, 821 09
Slovakia

Email: ppsenak@cisco.com

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

Email: acee@cisco.com

Les Ginsberg
Cisco Systems
821 Alder Drive
MILPITAS, CA 95035
USA

Email: ginsberg@cisco.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp, 2018 94089
Belgium

Email: wim.henderickx@nokia.com

Jeff Tantsura
Individual
USA

Email: jefftant.ietf@gmail.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com