

Internet
Internet-Draft
Intended status: Informational
Expires: September 2, 2015

D. Yeung
Y. Qu
Cisco Systems
J. Zhang
D. Bogdanovic
Juniper Networks
K. Sreenivasa
Brocade Communications System
March 2015

Yang Data Model for OSPF Protocol
draft-ietf-ospf-yang-01

Abstract

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on September 2, 2015.

Copyright Notice

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

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

Internet-Draft

OSPF Yang Data Model

March 2015

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
2.	Design of Data Model	3
2.1.	Overview	3
2.2.	OSPFv2 and OSPFv3	5
2.3.	Optional Features	5
2.4.	Inheritance	5
2.5.	OSPF Router Configuration	5
2.6.	OSPF Instance Configuration	6
2.7.	OSPF Area Configuration	7
2.8.	OSPF Interface Configuration	9
2.9.	OSPF notification	11
3.	OSPF Segment Routing	14
4.	OSPF Yang Module	20
5.	OSPF Segment Routing Yang Module	90
6.	Security Considerations	104
7.	Acknowledgements	104
8.	References	104
8.1.	Normative References	104
8.2.	Informative References	105
	Authors' Addresses	105

[1.](#) Overview

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

A core routing data model is defined in [[I-D.ietf-netmod-routing-cfg](#)], and it proposes a basis for the development of data models for routing protocols. The interface data model is defined in [[RFC7223](#)] and is used for referencing interface from the routing protocol. The key-chain data model is defined in

[\[I-D.acee-rtg-yang-key-chain\]](#) and is used for referencing key-chains configured for authentication and for the enumeration of cryptographic algorithms, also used for authentication. This document defines a YANG data model that can be used to configure and manage OSPF and it is an augment to the core routing data model.

This document defines a YANG data model that can be used to configure and manage OSPF. Both OSPFv2 [\[RFC2328\]](#) and OSPFv3 [\[RFC5340\]](#) are supported. In addition to the core OSPF protocol, features described in different separate OSPF RFCs are also supported. They includes demand circuit [\[RFC1793\]](#), traffic engineering [\[RFC3630\]](#), multiple address family [\[RFC5838\]](#), graceful restart [\[RFC3623\]](#) [\[RFC5187\]](#), NSSA [\[RFC3101\]](#) and sham link [\[RFC4577\]](#). Those non-core features are made optional in the data model provided.

[1.1.](#) Requirements Language

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

[2.](#) Design of Data Model

Although the basis of OSPF configuration elements like routers, areas and interfaces remains the same, the detailed configuration model varies among different vendors. Differences are observed in term of how protocol engine is tied to routing domain, how multiple protocol engines could be instantiated and configuration inheritance, just to name a few.

The goal of this document is to define a data model that is capable of representing these differences. There is very little information that is designated as "mandatory", providing freedom to vendors to adapt this data model to their product implementation.

[2.1.](#) Overview

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

The OSPF YANG module augments the routing/routing-instance/routing-protocols/routing-protocol path of the ietf-routing module.

```
module: ospf
+--rw routing
+--rw routing-instance [name]
+--rw routing-protocols
+--rw routing-protocol [name]
+--rw ospf
    .
    .
+--rw all-instances-inherit {instance-inheritance}?
|   .
|   .
+--rw instance* [routing-instance af]
    .
    .
+--rw areas
|   +--rw area* [area-id]
|       .
|       .
|       +--rw all-interfaces-inherit {interface-inheritance}?
|       .
|       .
|       +--rw virtual-links
|           +--rw virtual-link* [router-id]
|               .
|               .
|       +--rw sham-links
|           +--rw sham-link* [local-id remote-id]
|               .
|               .
```

```

|      +--rw interfaces
|      |      +--rw interface* [interface]
|      |      .
|      |      .
|      .
|      .
+--rw topologies
      +--rw topology* [name]

```

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

The ospf container includes one or more OSPF protocol engines, each encapsulated in the instance entity. Each instance includes information for the routing domain it is running on based on the [routing-instance af] specification. There is no default routing domain assumed by the data model. For example, to enable OSPF on the default IPv4 routing domain of the vendor, this model requires an

explicit instance entity with the specification like ["default" "ipv4-unicast"]. The instance also contains OSPF router level configuration

The instance/area and instance/area/interface container contain the OSPF configuration for the area and interface level respectively

The instance/topology container contain the OSPF configuration for topology when multi-topology feature is enabled

[2.2.](#) OSPFv2 and OSPFv3

The defined data model supports both OSPFv2 and OSPFv3.

The field 'version' is used to indicate the OSPF version and is a mandatory. Based on the version set, the data model change accordingly to accommodate the difference between the two versions.

[2.3.](#) Optional Features

Optional features are features beyond the basic of OSPF configurations and it is up to a vendor to decide the support of a

particular feature on a particular device.

This module has declared a number of features, such as NSR, max-LSA etc.. It is intended that vendors will extend the features list.

[2.4.](#) Inheritance

This defined data model supports configuration inheritance for instances, areas and interfaces.

The all-instances-inherit, all-areas-inherit and all-interfaces-inherit containers provides a consistent way to configure inheritable command. Inheritance is treated as a feature. Vendors are expected to augment the above container to provide the list of inheritance command for their implementation.

[2.5.](#) OSPF Router Configuration

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

```
module: ospf
  +--rw ospf
    +--rw all-instances-inherit {instance-inheritance}?
      | +--rw area
      | +--rw interface
    +--rw operation-mode?          identityref
    +--rw instance* [af]
      .
      .
```

[2.6.](#) OSPF Instance Configuration

The container instance represents an OSPF protocol engine. Each instance indicates the routing domain it is associated with based on [routing-instance af] and contains the router level configurations.

The all-areas-inherit container contains area configuration that could be inherited to all OSPF areas defined. Similarly, the all-areas-inherit also contains interface configuration that could be inherited to all the OSPF interfaces defined.

module: ospf

```

+--rw ospf
  .
  .
+--rw instance* [af]
  +--rw af identityref
  +--rw router-id? yang:dotted-quad {router-id}?
  +--rw admin-distance
    | +--rw (scope)?
    |   +--:(single-value)
    |   | +--rw all? uint8
    |   +--:(multi-values)
    |   +--rw (granularity)?
    |   | +--:(detail)
    |   | | +--rw intra-area? uint8
    |   | | +--rw inter-area? uint8
    |   | +--:(coarse)
    |   | +--rw internal? uint8
    |   +--rw external? uint8
+--rw nsr {nsr}?
  | +--rw enable? boolean
+--rw graceful-restart {graceful-restart}?
  | +--rw enable? boolean
  | +--rw helper-enable? boolean
  | +--rw restart-interval? uint16
  | +--rw helper-strict-lsa-checking? boolean

```

```

+--rw enable? boolean {admin-control}?
+--rw auto-cost {auto-cost}?
  | +--rw enable? boolean
  | +--rw reference-bandwidth? uint32
+--rw spf-control
  | +--rw paths? uint16 {max-ecmp}?
+--rw 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 (source)?
|       |       +--:(interface)
|       |           |   +--rw interface?   if:interface-ref
|       |           +--:(explicit)
|       |               +--rw router-id?   inet:ipv4-address
|   +--rw ldp
|       +--rw igp-sync?       boolean {ldp-igp-sync}?
|       +--rw autoconfig?    boolean {ldp-igp-autoconfig}?
+--rw fast-reroute {fast-reroute}?
|   +--rw lfa {lfa}?
+--rw all-areas-inherit {area-inheritance}?
|   +--rw area
|       +--rw interface

```

2.7. OSPF Area Configuration

The container area contains configurations of that area and the list of interface container represents all the OSPF interfaces active in the enclosing area.

```

module: ospf
+--rw ospf
.
.
+--rw instance* [routing-instance af]
.
.
+--rw areas
|   +--rw area* [area-id]
|       +--rw area-id                area-id-type
|       +--rw area-type?             identityref
|       +--rw summary?               boolean
|       +--rw default-cost?          uint32
|       +--rw ranges

```

```

|   |   +--rw range* [prefix]

```



```

|         +---rw prefix          inet:ip-prefix
|         +---rw advertise?      boolean
|         +---rw cost?          uint24
+---rw all-interfaces-inherit {interface-inheritance}?
|   +---rw interface
+---rw virtual-links
|   +---rw virtual-link* [router-id]
|     +---rw router-id          yang:dotted-quad
|     +---rw cost?              uint16
|     +---rw hello-interval?    uint16
|     +---rw dead-interval?     uint16
|     +---rw retransmit-interval? uint16
|     +---rw transmit-delay?    uint16
|     +---rw mtu-ignore?        boolean {mtu-ignore}?
|     +---rw lls?               boolean {lls}?
|     +---rw prefix-suppression? boolean {prefix-suppression}?
|     +---rw bfd?               boolean {bfd}?
|     +---rw ttl-security {ttl-security}?
|       +---rw enable?          boolean
|       +---rw hops?            uint8
+---rw enable?                  boolean {admin-control}?
+---rw authentication
|   +---rw (auth-type-selection)?
|     +---:(auth-ipsec) {ospfv3-authentication-ipsec}?
|       +---rw sa?              string
|     +---:(auth-trailer-key-chain)
|       +---rw key-chain?        key-chain:key-chain-ref
|     +---:(auth-trailer-key)
|       +---rw key?              string
|       +---rw crypto-algorithm
|         +---rw (algorithm)?
|           +---:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
|             +---rw hmac-sha1-12? empty
|           +---:(md5)
|             +---rw md5?          empty
|           +---:(sha-1)
|             +---rw sha-1?        empty
|           +---:(hmac-sha-1)
|             +---rw hmac-sha-1?   empty
|           +---:(hmac-sha-256)
|             +---rw hmac-sha-256? empty
|           +---:(hmac-sha-384)
|             +---rw hmac-sha-384? empty
|           +---:(hmac-sha-512)
|             +---rw hmac-sha-512? empty
+---rw sham-link
|   +---rw sham-link* [local-id remote-id]

```

```

+--rw local-id                inet:ip-address
+--rw remote-id               inet:ip-address
+--rw cost?                   uint16
+--rw hello-interval?        uint16
+--rw dead-interval?         uint16
+--rw retransmit-interval?    uint16
+--rw transmit-delay?        uint16
+--rw mtu-ignore?             boolean {mtu-ignore}?
+--rw lls?                    boolean {lls}?
+--rw prefix-suppression?    boolean {prefix-suppression}?
+--rw bfd?                    boolean {bfd}?
+--rw ttl-security {ttl-security}?
|   +--rw enable?             boolean
|   +--rw hops?               uint8
+--rw enable?                  boolean {admin-control}?
+--rw authentication
|   +--rw (auth-type-selection)?
|   |   +--:(auth-ipsec) {ospfv3-authentication-ipsec}?
|   |   |   +--rw sa?                string
|   |   +--:(auth-trailer-key-chain)
|   |   |   +--rw key-chain?          key-chain:key-chain-ref
|   |   +--:(auth-trailer-key)
|   |   |   +--rw key?                string
|   |   +--rw crypto-algorithm
|   |   |   +--rw (algorithm)?
|   |   |   |   +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
|   |   |   |   |   +--rw hmac-sha1-12?    empty
|   |   |   |   +--:(md5)
|   |   |   |   |   +--rw md5?              empty
|   |   |   |   +--:(sha-1)
|   |   |   |   |   +--rw sha-1?            empty
|   |   |   |   +--:(hmac-sha-1)
|   |   |   |   |   +--rw hmac-sha-1?      empty
|   |   |   |   +--:(hmac-sha-256)
|   |   |   |   |   +--rw hmac-sha-256?    empty
|   |   |   |   +--:(hmac-sha-384)
|   |   |   |   |   +--rw hmac-sha-384?    empty
|   |   |   |   +--:(hmac-sha-512)
|   |   |   |   |   +--rw hmac-sha-512?    empty

```

[2.8.](#) OSPF Interface Configuration

The container interface contains configurations of that interface.

The ospf-interfaces also contain interface configuration that could be inherited to all ospf-interface's defined.

```

+--rw ospf
  .
  .
+--rw instance* [routing-instance af]
  .
  .
+--rw areas
  | +--rw area* [area-id]
  |   .
  |   .
  |   +--rw interfaces
  |   | +--rw interface* [interface]
  |   |   +--rw interface          if:interface-ref
  |   |   +--rw network-type?      enumeration
  |   |   +--rw passive?           boolean
  |   |   +--rw demand-circuit?    boolean {demand-circuit}?
  |   |   +--rw multi-areas
  |   |   | +--rw multi-area* [multi-area-id] {multi-area-adj}?
  |   |   |   +--rw multi-area-id  area-id-type
  |   |   |   +--rw cost?           uint16
  |   |   +--rw static-neighbors
  |   |   | +--rw neighbor* [address]
  |   |   |   +--rw address         inet:ip-address
  |   |   |   +--rw cost?           uint16
  |   |   |   +--rw poll-interval?  uint16
  |   |   |   +--rw priority?       uint8
  |   |   +--rw node-flag?          boolean {node-flag}?
  |   |   +--rw fast-reroute {fast-reroute}?
  |   |   | +--rw lfa {lfa}?
  |   |   |   +--rw candidate-disabled?  boolean
  |   |   |   +--rw enabled?             boolean
  |   |   |   +--rw remote-lfa {remote-lfa}?
  |   |   |   +--rw enabled?             boolean
  |   |   +--rw cost?                 uint16
  |   |   +--rw hello-interval?       uint16
  |   |   +--rw dead-interval?         uint16
  |   |   +--rw retransmit-interval?   uint16
  |   |   +--rw transmit-delay?        uint16
  |   |   +--rw mtu-ignore?            boolean {mtu-ignore}?

```

```

|      +---rw lls?                               boolean {lls}?
|      +---rw prefix-suppression?                boolean {prefix-suppression}?
|      +---rw bfd?                               boolean {bfd}?
|      +---rw ttl-security {ttl-security}?
|      |   +---rw enable?                        boolean
|      |   +---rw hops?                          uint8
|      +---rw enable?                            boolean {admin-control}?
|      +---rw authentication
|      |   +---rw (auth-type-selection)?

```

```

|      |      +---:(auth-ipsec) {ospfv3-authentication-ipsec}?
|      |      |   +---rw sa?                               string
|      |      +---:(auth-trailer-key-chain)
|      |      |   +---rw key-chain?                       key-chain:key-chain-ref
|      |      +---:(auth-trailer-key)
|      |      |   +---rw key?                             string
|      |      |   +---rw crypto-algorithm
|      |      |   |   +---rw (algorithm)?
|      |      |   |   |   +---:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
|      |      |   |   |   |   +---rw hmac-sha1-12?      empty
|      |      |   |   +---:(md5)
|      |      |   |   |   +---rw md5?                    empty
|      |      |   |   +---:(sha-1)
|      |      |   |   |   +---rw sha-1?                  empty
|      |      |   |   +---:(hmac-sha-1)
|      |      |   |   |   +---rw hmac-sha-1?             empty
|      |      |   |   +---:(hmac-sha-256)
|      |      |   |   |   +---rw hmac-sha-256?          empty
|      |      |   |   +---:(hmac-sha-384)
|      |      |   |   |   +---rw hmac-sha-384?          empty
|      |      |   |   +---:(hmac-sha-512)
|      |      |   |   |   +---rw hmac-sha-512?          empty
|      |      +---rw topologies
|      |      |   +---rw topology* [name]
|      |      |   |   +---rw name                        leafref
|      |      |   |   +---rw cost?                       uint32

```

[2.9.](#) OSPF notification

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the common set of traps from OSPFv2 MIB

[RFC4750] and OSPFv3 MIB [RFC5643].

```
module: ospf
notifications:
  +---n if-state-change
  |   +--ro routing-instance?      rt:routing-instance-ref
  |   +--ro routing-protocol-type? leafref
  |   +--ro routing-protocol-name? leafref
  |   +--ro af?                    leafref
  |   +--ro link-type?             identityref
  |   +--ro interface
  |   |   +--ro interface?        if:interface-ref
  |   +--ro virtual-link
  |   |   +--ro area-id?          uint32
  |   |   +--ro neighbor-router-id? yang:dotted-quad
  |   +--ro sham-link
```

```
  |   |   +--ro area-id?          uint32
  |   |   +--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-instance?      rt:routing-instance-ref
  |   +--ro routing-protocol-type? leafref
  |   +--ro routing-protocol-name? leafref
  |   +--ro af?                    leafref
  |   +--ro link-type?             identityref
  |   +--ro interface
  |   |   +--ro interface?        if:interface-ref
  |   |   +--ro packet-source?    yang:dotted-quad
  |   +--ro virtual-link
  |   |   +--ro area-id?          uint32
  |   |   +--ro neighbor-router-id? yang:dotted-quad
  |   +--ro sham-link
  |   |   +--ro area-id?          uint32
  |   |   +--ro local-ip-addr?    inet:ip-address
  |   |   +--ro remote-ip-addr?   inet:ip-address
  |   +--ro packet-type?          packet-type
  |   +--ro error?                 enumeration
  +---n nbr-state-change
  |   +--ro routing-instance?      rt:routing-instance-ref
  |   +--ro routing-protocol-type? leafref
```

```

| +--ro routing-protocol-name? leafref
| +--ro af? leafref
| +--ro link-type? identityref
| +--ro interface
| | +--ro interface? if:interface-ref
| | +--ro neighbor-router-id? yang:dotted-quad
| | +--ro neighbor-ip-addr? yang:dotted-quad
| +--ro virtual-link
| | +--ro area-id? uint32
| | +--ro neighbor-router-id? yang:dotted-quad
| +--ro sham-link
| | +--ro area-id? uint32
| | +--ro local-ip-addr? inet:ip-address
| | +--ro neighbor-router-id? yang:dotted-quad
| | +--ro neighbor-ip-addr? yang:dotted-quad
| +--ro state? nbr-state-type
+---n nbr-restart-helper-status-change
| +--ro routing-instance? rt:routing-instance-ref
| +--ro routing-protocol-type? leafref
| +--ro routing-protocol-name? leafref
| +--ro af? leafref
| +--ro link-type? identityref
| +--ro interface

```

```

| | +--ro interface? if:interface-ref
| | +--ro neighbor-router-id? yang:dotted-quad
| | +--ro neighbor-ip-addr? yang:dotted-quad
| +--ro virtual-link
| | +--ro area-id? uint32
| | +--ro neighbor-router-id? yang:dotted-quad
| +--ro status? restart-helper-status-type
| +--ro age? uint32
| +--ro exit-reason? restart-exit-reason-type
+---n rx-bad-packet
| +--ro routing-instance? rt:routing-instance-ref
| +--ro routing-protocol-type? leafref
| +--ro routing-protocol-name? leafref
| +--ro af? leafref
| +--ro link-type? identityref
| +--ro interface
| | +--ro interface? if:interface-ref
| | +--ro packet-source? yang:dotted-quad

```

```

|   +---ro virtual-link
|   |   +---ro area-id?          uint32
|   |   +---ro neighbor-router-id?  yang:dotted-quad
|   +---ro sham-link
|   |   +---ro area-id?          uint32
|   |   +---ro local-ip-addr?    inet:ip-address
|   |   +---ro remote-ip-addr?   inet:ip-address
|   +---ro packet-type?          packet-type
+---n lsdb-approaching-overflow
|   +---ro routing-instance?      rt:routing-instance-ref
|   +---ro routing-protocol-type? leafref
|   +---ro routing-protocol-name? leafref
|   +---ro af?                    leafref
|   +---ro ext-lsdb-limit?        uint32
+---n lsdb-overflow
|   +---ro routing-instance?      rt:routing-instance-ref
|   +---ro routing-protocol-type? leafref
|   +---ro routing-protocol-name? leafref
|   +---ro af?                    leafref
|   +---ro ext-lsdb-limit?        uint32
+---n nssa-translator-status-change
|   +---ro routing-instance?      rt:routing-instance-ref
|   +---ro routing-protocol-type? leafref
|   +---ro routing-protocol-name? leafref
|   +---ro af?                    leafref
|   +---ro area-id?              uint32
|   +---ro status?               nssa-translator-state-type
+---n restart-status-change
|   +---ro routing-instance?      rt:routing-instance-ref
|   +---ro routing-protocol-type? leafref

```

```

+---ro routing-protocol-name?  leafref
+---ro af?                      leafref
+---ro status?                  restart-status-type
+---ro restart-interval?       uint16
+---ro exit-reason?             restart-exit-reason-type

```

3. OSPF Segment Routing

In addition to the OSPF base YANG model, this document also defines a model for OSPF segment routing.

The OSPF SR YANG module requires the base segment routing module [[I-D.litkowski-spring-sr-yang](#)] to be supported as there is a strong relationship between those modules.

```
module: ietf-ospf-sr
augment /rt:routing/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance:
    +--rw segment-routing
        +--rw enabled?    boolean
        +--rw bindings
            +--rw advertise
                | +--rw policies*    string
            +--rw receive?    boolean
augment /rt:routing/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:interfaces/ospf:interface:
    +--rw segment-routing
        +--rw adjacency-sid
            +--rw advertise-adj-group-sid* [group-id]
                | +--rw group-id    uint32
            +--rw advertise-protection?    enumeration
augment /rt:routing/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:interfaces/ospf:interface/ospf:fast-reroute:
    +--rw ti-lfa {ti-lfa}?
        +--rw enable?    boolean
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance:
    +--ro segment-routing
        +--ro enabled?    boolean
        +--ro bindings
            +--ro advertise
                | +--ro policies*    string
            +--ro receive?    boolean
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:interfaces/ospf:interface:
```

```
+--ro segment-routing
    +--ro adjacency-sid
        +--ro advertise-adj-group-sid* [group-id]
            | +--ro group-id    uint32
```



```

        +---ro advertise-protection?      enumeration
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
  rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:interfaces/ospf:interface/ospf:database/
      ospf:link-scope-lsa-type/ospf:link-scope-lsas/
        ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2/
          ospf:body/ospf:opaque/ospf:extended-prefix-tlvs/
            ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
| +---ro prefix-sid-sub-tlv*
|   +---ro flags?      bits
|   +---ro mt-id?      uint8
|   +---ro algorithm?  uint8
|   +---ro sid?        uint32
+---ro sid-binding-sub-tlvs
  +---ro sid-binding-sub-tlv*
    +---ro flags?      bits
    +---ro mt-id?      uint8
    +---ro weight?     uint8
    +---ro sid-sub-tlv
    | +---ro sid?      uint32
    +---ro ero-metric-sub-tlv
    | +---ro metric?   uint32
    +---ro ipv4-ero-sub-tlv
    | +---ro flags?    bits
    | +---ro ipv4-address?  inet:ipv4-address
    +---ro unnumbered-ero-sub-tlv
    | +---ro flags?    bits
    | +---ro router-id?  yang:dotted-quad
    | +---ro interface-id? uint32
    +---ro ipv4-backup-ero-sub-tlv
    | +---ro flags?    bits
    | +---ro ipv4-address?  inet:ipv4-address
    +---ro unnumbered-backup-ero-sub-tlv
    +---ro flags?      bits
    +---ro router-id?  yang:dotted-quad
    +---ro interface-id? uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
  rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:database/ospf:area-scope-lsa-type/
      ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version/
        ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque/
          ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs

```

```

|   +---ro prefix-sid-sub-tlv*
|       +---ro flags?          bits
|       +---ro mt-id?          uint8
|       +---ro algorithm?      uint8
|       +---ro sid?            uint32
+---ro sid-binding-sub-tlvs
    +---ro sid-binding-sub-tlv*
        +---ro flags?          bits
        +---ro mt-id?          uint8
        +---ro weight?         uint8
        +---ro sid-sub-tlv
            | +---ro sid?      uint32
        +---ro ero-metric-sub-tlv
            | +---ro metric?   uint32
        +---ro ipv4-ero-sub-tlv
            | +---ro flags?     bits
            | +---ro ipv4-address?  inet:ipv4-address
        +---ro unnumbered-ero-sub-tlv
            | +---ro flags?     bits
            | +---ro router-id?  yang:dotted-quad
            | +---ro interface-id?  uint32
        +---ro ipv4-backup-ero-sub-tlv
            | +---ro flags?     bits
            | +---ro ipv4-address?  inet:ipv4-address
        +---ro unnumbered-backup-ero-sub-tlv
            +---ro flags?         bits
            +---ro router-id?     yang:dotted-quad
            +---ro interface-id?  uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
    rt:routing-protocol/ospf:ospf/ospf:instance/ospf:database/
    ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa/
    ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque/
    ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
|   +---ro prefix-sid-sub-tlv*
|       +---ro flags?          bits
|       +---ro mt-id?          uint8
|       +---ro algorithm?      uint8
|       +---ro sid?            uint32
+---ro sid-binding-sub-tlvs
    +---ro sid-binding-sub-tlv*
        +---ro flags?          bits
        +---ro mt-id?          uint8
        +---ro weight?         uint8
        +---ro sid-sub-tlv
            | +---ro sid?      uint32
        +---ro ero-metric-sub-tlv

```

| +--ro metric? uint32

```
+--ro ipv4-ero-sub-tlv
| +--ro flags? bits
| +--ro ipv4-address? inet:ipv4-address
+--ro unnumbered-ero-sub-tlv
| +--ro flags? bits
| +--ro router-id? yang:dotted-quad
| +--ro interface-id? uint32
+--ro ipv4-backup-ero-sub-tlv
| +--ro flags? bits
| +--ro ipv4-address? inet:ipv4-address
+--ro unnumbered-backup-ero-sub-tlv
  +--ro flags? bits
  +--ro router-id? yang:dotted-quad
  +--ro interface-id? uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
  rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:database/ospf:area-scope-lsa-type/
      ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version/
        ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque/
          ospf:extended-link-tlvs/ospf:extended-link-tlv:
+--ro adj-sid-sub-tlvs
| +--ro adj-sid-sub-tlv*
|   +--ro flags? bits
|   +--ro mt-id? uint8
|   +--ro weight? uint8
|   +--ro sid? uint32
+--ro lan-adj-sid-sub-tlvs
  +--ro lan-adj-sid-sub-tlv*
    +--ro flags? bits
    +--ro mt-id? uint8
    +--ro weight? uint8
    +--ro neighbor-router-id? yang:dotted-quad
    +--ro sid? uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
  rt:routing-protocol/ospf:ospf/ospf:instance/
    ospf:areas/ospf:area/ospf:interfaces/ospf:interface/
      ospf:database/ospf:link-scope-lsa-type/
        ospf:link-scope-lsas/ospf:link-scope-lsa/ospf:version/
          ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque:
+--ro extended-prefix-range-tlvs
```

```

|   +---ro extended-prefix-range-tlv*
|       +---ro range-size?          uint16
|       +---ro flags?               bits
|       +---ro prefix?              inet:ip-prefix
|       +---ro perfix-sid-sub-tlvs
|           |   +---ro prefix-sid-sub-tlv*
|           |       +---ro flags?      bits
|           |       +---ro mt-id?      uint8

```

```

|       |   +---ro algorithm?    uint8
|       |   +---ro sid?          uint32
|   +---ro sid-binding-sub-tlvs
|       |   +---ro sid-binding-sub-tlv*
|       |       +---ro flags?          bits
|       |       +---ro mt-id?          uint8
|       |       +---ro weight?         uint8
|       |       +---ro sid-sub-tlv
|       |           |   +---ro sid?    uint32
|       |       +---ro ero-metric-sub-tlv
|       |           |   +---ro metric?  uint32
|       |       +---ro ipv4-ero-sub-tlv
|       |           |   +---ro flags?    bits
|       |           |   +---ro ipv4-address?  inet:ipv4-address
|       |       +---ro unnumbered-ero-sub-tlv
|       |           |   +---ro flags?    bits
|       |           |   +---ro router-id?  yang:dotted-quad
|       |           |   +---ro interface-id?  uint32
|       |       +---ro ipv4-backup-ero-sub-tlv
|       |           |   +---ro flags?    bits
|       |           |   +---ro ipv4-address?  inet:ipv4-address
|       |       +---ro unnumbered-backup-ero-sub-tlv
|       |           |   +---ro flags?    bits
|       |           |   +---ro router-id?  yang:dotted-quad
|       |           |   +---ro interface-id?  uint32
|   +---ro unknown-tlvs
|       +---ro unknown-tlv*
|           +---ro type?      uint16
|           +---ro length?    uint16
|           +---ro value?     yang:hex-string
+---ro sr-algorithm-tlv
|   +---ro sr-algorithm*    uint8
+---ro sid-range-tlvs

```

```

    +--ro sid-range-tlv*
      +--ro range-size?    ospf:uint24
      +--ro sid-sub-tlv
        +--ro sid?    uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
  rt:routing-protocol/ospf:ospf/ospf:instance/ospf:areas/
    ospf:area/ospf:database/ospf:area-scope-lsa-type/
      ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version/
        ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque:
  +--ro extended-prefix-range-tlvs
    | +--ro extended-prefix-range-tlv*
    |   +--ro range-size?          uint16
    |   +--ro flags?              bits
    |   +--ro prefix?             inet:ip-prefix
    |   +--ro prefix-sid-sub-tlvs

```

```

    |   +--ro prefix-sid-sub-tlv*
    |     +--ro flags?          bits
    |     +--ro mt-id?          uint8
    |     +--ro algorithm?      uint8
    |     +--ro sid?            uint32
    |   +--ro sid-binding-sub-tlvs
    |     +--ro sid-binding-sub-tlv*
    |       +--ro flags?              bits
    |       +--ro mt-id?              uint8
    |       +--ro weight?             uint8
    |       +--ro sid-sub-tlv
    |         | +--ro sid?    uint32
    |       +--ro ero-metric-sub-tlv
    |         | +--ro metric?  uint32
    |       +--ro ipv4-ero-sub-tlv
    |         | +--ro flags?          bits
    |         | +--ro ipv4-address?  inet:ipv4-address
    |       +--ro unnumbered-ero-sub-tlv
    |         | +--ro flags?          bits
    |         | +--ro router-id?      yang:dotted-quad
    |         | +--ro interface-id?   uint32
    |       +--ro ipv4-backup-ero-sub-tlv
    |         | +--ro flags?          bits
    |         | +--ro ipv4-address?  inet:ipv4-address
    |       +--ro unnumbered-backup-ero-sub-tlv
    |         +--ro flags?          bits

```

```

|         |         +---ro router-id?      yang:dotted-quad
|         |         +---ro interface-id?   uint32
|         +---ro unknown-tlvs
|             +---ro unknown-tlv*
|                 +---ro type?      uint16
|                 +---ro length?    uint16
|                 +---ro value?     yang:hex-string
+---ro sr-algorithm-tlv
| +---ro sr-algorithm*   uint8
+---ro sid-range-tlvs
    +---ro sid-range-tlv*
        +---ro range-size?    ospf:uint24
        +---ro sid-sub-tlv
            +---ro sid?      uint32
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/'
    rt:routing-protocol/ospf:ospf/ospf:instance/ospf:database/
    ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa/
    ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque:
+---ro extended-prefix-range-tlvs
| +---ro extended-prefix-range-tlv*
|     +---ro range-size?          uint16
|     +---ro flags?              bits

```

```

|         +---ro prefix?          inet:ip-prefix
|         +---ro prefix-sid-sub-tlvs
|             +---ro prefix-sid-sub-tlv*
|                 +---ro flags?      bits
|                 +---ro mt-id?      uint8
|                 +---ro algorithm?  uint8
|                 +---ro sid?        uint32
+---ro sid-binding-sub-tlvs
    +---ro sid-binding-sub-tlv*
        +---ro flags?              bits
        +---ro mt-id?              uint8
        +---ro weight?             uint8
        +---ro sid-sub-tlv
            +---ro sid?      uint32
+---ro ero-metric-sub-tlv
| +---ro metric?    uint32
+---ro ipv4-ero-sub-tlv
| +---ro flags?      bits
| +---ro ipv4-address?  inet:ipv4-address

```



```

}

import ietf-routing {
    prefix "rt";
}

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

organization
    "Cisco Systems
    170 West Tasman Drive
    San Jose, CA 95134-1706
    USA";

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

    WG Chair: Acee Lindem
               <mailto:acee@cisco.com>

    WG Chair: Abhay Roy
               <mailto:akr@cisco.com>

    Editor:    Derek Yeung
               <mailto:myeung@cisco.com>
    Author:    Derek Yeung
               <mailto:myeung@cisco.com>
    Author:    Yingzhen Qu
               <mailto:yiqu@cisco.com>
    Author:    Jeffrey Zhang
               <mailto:zzhang@juniper.net>
    Author:    Dean Bogdanovic
               <mailto:deanb@juniper.net>

```

```

    Author:    Kiran Agrahara Sreenivasa
               <mailto:kkoushik@Brocade.com>";

```

```

description
    "This YANG module defines the generic configuration

```


data for OSPF, which is common across all of the vendor implementations of the protocol. 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.

Terms and Acronyms

OSPF (ospf): Open Shortest Path First

IP (ip): Internet Protocol

IPv4 (ipv4): Internet Protocol Version 4

IPv6 (ipv6): Internet Protocol Version 6

MTU (mtu) Maximum Transmission Unit

";

revision 2015-07-06 {

description

- * Remove support for protocol-centric config.
- * Enclose list in container, except for instance.
- * Replace protocol-shutdown with admin-control.
- * Add IPFRR per-interface config.
- * Reorganize max-path etc node.
- * Add node-flag.
- * Align config/operation hierarchy.
- * Use relative path for reference to rib.
- * Add ability to set single admin distance.
- * Make unreserved bandwidth into list.
- * Add F and T bit to OSPFv3 external LSA.
- * Remove key statement inside LSA body.
- * Add stub router support.
- * Fix usage of af-area-config.
- * Add statistics to operation data.
- * Add local rib.
- * Use dotted-quad for all router-id fields.
- * Support more than one multi-area per interface.
- * Use uint16 for LSA type.
- * Update grouping notification-instance-hdr.
- * Rework condition for opaque type and id in OSPFv2 LSA.
- * Rename local-remote-ipv4-addr with remote-if-ipv4-addr.

```

        * Add virtual-link/sham-link to operation state.
        * Allow multiple link TLVs in one LSA.
        * Fix bug in as-scope-lsas.
        * Remove OSPFv3 restriction in link-scope-lsas.
        * Editorial changes.";
    reference
        "RFC XXXX: A YANG Data Model for OSPF";
}

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

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

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

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

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

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

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

identity normal {

```

Internet-Draft

OSPF Yang Data Model

March 2015

```
    base area-type;
    description "OSPF normal area.";
}

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

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

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

typedef area-id-type {
    type union {
        type uint32;
        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 external route type 1.";
        }
    }
}
```

```

enum external-2 {
    description "OSPF External route type 2.";
}
enum nssa-1 {
    description "OSPF NSSA external route type 1.";
}

```

```

enum nssa-2 {
    description "OSPF NSSA external route type 2.";
}
}
description "OSPF route type.";
}

typedef if-state-type {
    type enumeration {
        enum Down {
            value "1";
            description
                "Interface down state.";
        }
        enum Loopback {
            value "2";
            description
                "Interface loopback state.";
        }
        enum Waiting {
            value "3";
            description
                "Interface waiting state.";
        }
        enum Point-to-Point {
            value "4";
            description
                "Interface point-to-point state.";
        }
        enum DR {
            value "5";
            description
                "Interface Designated Router (DR) state.";
        }
        enum BDR {

```

```

        value "6";
        description
            "Interface Backup Designated Router (BDR) state.";
    }
    enum DR-Other {
        value "7";
        description
            "Interface Other Designated Router state.";
    }
}
description
    "OSPF interface state type.";
}

```

```

typedef nbr-state-type {
    type enumeration {
        enum Down {
            value "1";
            description
                "Neighbor down state.";
        }
        enum Attempt {
            value "2";
            description
                "Neighbor attempt state.";
        }
        enum Init {
            value "3";
            description
                "Neighbor init state.";
        }
        enum 2-Way {
            value "4";
            description
                "Neighbor 2-Way state.";
        }
        enum ExStart {
            value "5";
            description
                "Neighbor exchange start state.";
        }
        enum Exchange {

```

```

        value "6";
        description
            "Neighbor exchange state.";
    }
    enum Loading {
        value "7";
        description
            "Neighbor loading state.";
    }
    enum Full {
        value "8";
        description
            "Neighbor full state.";
    }
}
description
    "OSPF neighbor state type.";
}

typedef restart-helper-status-type {

```

```

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

typedef restart-exit-reason-type {
    type enumeration {
        enum None {
            value "1";
            description

```

```

        "Not attempted.";
    }
    enum InProgress {
        value "2";
        description
            "Restart in progress.";
    }
    enum Completed {
        value "3";
        description
            "Successfully completed.";
    }
    enum TimedOut {
        value "4";
        description
            "Timed out.";
    }
    enum TopologyChanged {
        value "5";
        description
            "Aborted due to topology change.";
    }
}
description
    "Describes the outcome of the last attempt at a
    graceful restart, either by itself or acting
    as a helper.";

```

```

}

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

```

```

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

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

```

```

        "NSSA translator disabled state.";
    }
}
description
    "OSPF NSSA translator state type.";
}

```



```

typedef restart-status-type {
  type enumeration {
    enum Not-Restarting {
      value "1";
      description
        "Router is not restarting.";
    }
    enum Planned-Restart {
      description
        "Router is going through planned restart.";
    }
    enum Unplanned-Restart {
      value "3";
      description
        "Router is going through unplanned restart.";
    }
  }
  description
    "OSPF graceful restart status type.";
}

feature multi-topology {
  description
    "Support MTR.";
}

feature multi-area-adj {
  description
    "OSPF multi-area adjacency support as in RFC 5185.";
}

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

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

```

```
feature mtu-ignore {
    description
        "Disable OSPF MTU mismatch detection on receiving
        DBD packets.";
}

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

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

feature bfd {
    description
        "OSPF BFD support.";
}

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

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

feature graceful-restart {
    description
        "Graceful OSPF Restart as defined in RFC3623 and RFC5187.";
}

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

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

feature max-ecmp {
```

Internet-Draft

OSPF Yang Data Model

March 2015

```
    description
      "Setting maximum number of ECMP paths.";
  }

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

  feature te-rid {
    description
      "TE router-id.";
  }

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

  feature ldp-igp-autoconfig {
    description
      "LDP IGP auto-config.";
  }

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

  feature fast-reroute {
    description
      "Support of IPFRR.";
  }

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

  feature lfa {
    description
      "Support of Loop Free Alternates.";
  }
```

```
feature remote-lfa {
  description
    "Support of remote Loop Free Alternates.";
}
```

```
feature stub-router {
  description
    "Support of RFC6987 OSPF Stub Router Advertisement";
}
```

```
feature instance-inheritance {
  description
    "Support instance inheritance";
}
```

```
feature area-inheritance {
  description
    "Support area inheritance";
}
```

```
feature interface-inheritance {
  description
    "Support interface inheritance";
}
```

```
grouping instance-stat {
  description "Per-instance statistics";
  leaf originate-new-lsa-count {
    type yang:counter32;
    description "The number of new LSAs originated.";
  }
  leaf rx-new-lsas-count {
    type yang:counter32;
    description "The number of LSAs received.";
  }
  leaf as-scope-lsa-count {
    type yang:gauge32;
    description "The number of as-scope LSAs.";
  }
  leaf as-scope-lsa-chksum-sum {
    type uint32;
  }
}
```

```

    description "The sum of the LS checksums.";
}
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 given type.";
    }
    leaf lsa-cksum-sum {
        type int32;
        description
            "The sum of the LS checksums of the given type.";
    }
}
}
}

grouping area-stat {
    description "Per-area statistics.";
    leaf spf-runs-count {
        type yang:counter32;
        description "The number of times that intra-area spf runs.";
    }
    leaf abr-count {
        type yang:gauge32;
        description "The total number of area border routers reachable
            within this area.";
    }
    leaf asbr-count {
        type yang:gauge32;
        description "The total number of AS border routers.";
    }
    leaf ar-nssa-translator-event-count {
        type yang:counter32;
        description "The number of translator state changes.";

```

```

}
leaf area-scope-lsa-count {
    type yang:gauge32;
    description "The number of LSAs in this area, excluding
                as-external LSAs.";
}
leaf area-scope-lsa-cksum-sum {
    type int32;
    description "The sum of the LSAs checksums.";
}
container database {
    description "Container for area scope LSA type statistics.";
    list area-scope-lsa-type {
        description "List of area scope LSA statistics";
        leaf lsa-type {
            type uint16;
            description "Area scope LSA type.";
        }
        leaf lsa-count {

```

```

    type yang:gauge32;
    description "The number of LSAs of the given type.";
}
leaf lsa-cksum-sum {
    type int32;
    description
        "The sum of the LS checksums of the given type.";
}
}
}
}

grouping interface-stat {
    description "Per-interface statistics";
    leaf if-event-count {
        type yang:counter32;
        description
            "The number of times this interface has changed its
             state or an error has occurred.";
    }
    leaf link-scope-lsa-count {
        type yang:gauge32;

```

```

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

```

```

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

```

```

grouping instance-fast-reroute-config {
  description
    "This group defines global configuration of IPFRR.";
  container fast-reroute {
    if-feature fast-reroute;
    description
      "This container may be augmented with global
        parameters for IPFRR.";
    container lfa {
      if-feature lfa;
      description
        "This container may be augmented with
          global parameters for LFA.
          Creating the container has no effect on
          LFA activation.";
    }
  }
}

grouping interface-fast-reroute-config {
  description
    "This group defines interface configuration of IPFRR.";
  container fast-reroute {
    if-feature fast-reroute;

    container lfa {
      if-feature lfa;

      leaf candidate-disabled {
        type boolean;
        description
          "Prevent the interface to be used as backup.";
      }
    }
  }
}

```

```

    leaf enabled {
      type boolean;
      description
        "Activates LFA.
        This model assumes activation of per-prefix LFA.";
    }
    container remote-lfa {
      if-feature remote-lfa;
    }
  }
}

```



```

        leaf enabled {
            type boolean;
            description
                "Activates remote LFA.";
        }
        description
            "Remote LFA configuration.";
    }
    description
        "LFA configuration.";
}
description
    "Fast-reroute configuration.";
}
}

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

    leaf cost {
        type uint16 {
            range "1..65535";
        }
        description
            "Interface cost.";
    }

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

    leaf dead-interval {
        type uint16 {
            range "1..65535";
        }
    }
}

```

}

```

units seconds;
must "../dead-interval > ../hello-interval" {
    error-message "The dead interval must be "
        + "larger than the hello interval";
    description
        "The value MUST be greater than 'hello-interval'.";
}
description
    "Interval after which a neighbor is declared dead.";
}

leaf retransmit-interval {
    type uint16 {
        range "1..65535";
    }
    units seconds;
    description
        "Time between retransmitting unacknowledged Link State
        Advertisements (LSAs).";
}

leaf transmit-delay {
    type uint16 {
        range "1..65535";
    }
    units seconds;
    description
        "Estimated time needed to send link-state update.";
}

leaf mtu-ignore {
    if-feature mtu-ignore;
    type boolean;
    description
        "Enable/Disable ignoring of MTU in DBD packets.";
}

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

leaf prefix-suppression {
    if-feature prefix-suppression;
    type boolean;

```

```
    description
      "Suppress advertisement of the prefixes.";
  }

  leaf bfd {
    if-feature bfd;
    type boolean;
    description
      "Enable/disable bfd.";
  }

  container ttl-security {
    if-feature ttl-security;
    description "TTL security check.";
    leaf enable {
      type boolean;
      description
        "Enable/Disable TTL security check.";
    }
    leaf hops {
      type uint8 {
        range "1..254";
      }
      description
        "Maximum number of hops that a OSPF packet may
        have traveled.";
    }
  }
}

leaf enable {
  if-feature admin-control;
  type boolean;
  default true;
  description
    "Enable/disable protocol on the interface.";
}

container authentication {
  description "Authentication configuration.";
  choice auth-type-selection {
    description
      "Options for expressing authentication setting.";
    case auth-ipsec {
      when "../..../rt:type = 'ospfv3'" {
        description "Applied to OSPFv3 only.";
      }
    }
    if-feature ospfv3-authentication-ipsec;
```

```
leaf sa {
    type string;
```

```
        description
            "SA name.";
    }
}
case auth-trailer-key-chain {
    leaf key-chain {
        type key-chain:key-chain-ref;
        description
            "key-chain name.";
    }
}
case auth-trailer-key {
    leaf key {
        type string;
        description
            "Key string in ASCII format.";
    }
    container crypto-algorithm {
        uses key-chain:crypto-algorithm-types;
        description
            "Cryptographic algorithm associated with key.";
    }
}
}
}
} // interface-common-config

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

    leaf network-type {
        type enumeration {
            enum "broadcast" {
                description
                    "Specify OSPF broadcast multi-access network.";
            }
            enum "non-broadcast" {
                description
                    "Specify OSPF Non-Broadcast Multi-Access
```

```

        (NBMA) network.";
    }
    enum "point-to-multipoint" {
        description
            "Specify OSPF point-to-multipoint network.";
    }
    enum "point-to-point" {
        description
            "Specify OSPF point-to-point network.";
    }

```

```

    }
}
description
    "Network type.";
}

leaf passive {
    type boolean;
    description
        "Enable/Disable passive.";
}

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

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

```

```

        type uint16;
        description
            "Interface cost for multi-area.";
    }
}

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

    list neighbor {
        key "address";
        description
            "Specify a neighbor router.";

        leaf address {

```

```

        type inet:ip-address;
        description "Neighbor IP address.";
    }

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

```

```

leaf node-flag {
    if-feature node-flag;
    type boolean;
    default false;
    description
        "Set prefix as a node representative prefix.";
}
uses interface-fast-reroute-config;
uses interface-common-config;
} // grouping interface-config

grouping tlv {
    description
        "TLV";
    leaf type {
        type uint16;
        description "TLV type.";
    }
    leaf length {
        type uint16;
        description "TLV length.";
    }
}

```

```

leaf value {
    type yang:hex-string;
    description "TLV value.";
}
}

grouping unknown-tlvs {
    description
        "Unknown TLVs grouping.
        Could be used for unknown TLVs or unknown sub-TLVs.";
    container unknown-tlvs {
        description "All unknown TLVs.";
        list unknown-tlv {
            description "Unknown TLV.";
            uses tlv;
        }
    }
}
}

```

```

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

```

```

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

```

```

description
  "Router LSA.";
leaf flags {
  type bits {
    bit V {
      description
        "When set, the router is an endpoint of one or
        more virtual links.";
    }
    bit E {
      description

```



```

        "When set, the router is an AS Boundary Router
        (ASBR).";
    }
    bit B {
        description
            "When set, the router is an Area Border
            Router (ABR).";
    }
}
description "Flags.";
}
leaf num-of-links {
    type uint16;
    description "Number of links.";
}
container links {
    description "All router Links.";
    list link {
        description "Router LSA link.";
        uses ospfv2-router-link;
        container topologies {
            description "All topologies for the link.";
            list topology {
                description
                    "Topology specific information.";
                leaf mt-id {
                    type uint8;
                    description
                        "The MT-ID for topology enabled on the link.";
                }
                leaf metric {
                    type uint16;
                    description "Metric for the topology.";
                }
            }
        }
    }
}
}

```

```

    }
}
container network {
    when "../header/type = 2" {

```

```

        description
            "Only apply to network LSA.";
    }
    description
        "Network LSA.";
    leaf network-mask {
        type inet:ipv4-address;
        description
            "The IP address mask for the network.";
    }
    container attached-routers {
        description "All attached routers.";
        leaf-list attached-router {
            type yang:dotted-quad;
            description
                "List of the routers attached to the network.";
        }
    }
}
container summary {
    when "../../header/type = 3 or "
        + "../../header/type = 4" {
        description
            "Only apply to Summary-LSA.";
    }
    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.";
        list topology {
            description
                "Topology specific information.";
            leaf mt-id {
                type uint8;
                description
                    "The MT-ID for topology enabled on the link.";
            }
            leaf metric {
                type uint24;
            }
        }
    }
}

```

```
        description "Metric for the topology.";
    }
}
}
container external {
    when "../../header/type = 5 or "
        + "../../header/type = 7" {
        description
            "Only apply to AS-external-LSA and NSSA-LSA.";
    }
    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 topology enabled on the link.";
            }
            leaf flags {
                type bits {
                    bit E {
                        description
                            "When set, the metric specified is a Type 2
                            external metric.";
                    }
                }
                description "Flags.";
            }
            leaf metric {
                type uint24;
                description "Metric for the topology.";
            }
            leaf forwarding-address {
                type inet:ipv4-address;
                description
                    "Forwarding address.";
            }
        }
    }
}
```

leaf external-route-tag {

```
        type uint32;
        description
            "Route tag.";
    }
}
}
}
container opaque {
    when "../../header/type = 9 or "
        + "../../header/type = 10 or "
        + "../../header/type = 11" {
        description
            "Only apply to opaque LSA.";
    }
    description
        "Opaque LSA.";

    uses unknown-tlvs;

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

container link-tlvs {
    description "All link TLVs in the LSA.";
    container link-tlv {
        description "Link TLV.";
        leaf link-type {
            type uint8;
            mandatory true;
            description "Link type.";
        }
        leaf link-id {
            type union {
```

```

        type inet:ipv4-address;
        type yang:dotted-quad;
    }
    mandatory true;
    description "Link ID.";
}
container local-if-ipv4-addr {
    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-addr {
    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 decimal64 {
        fraction-digits 2;
    }
    description "Maximum bandwidth.";
}
leaf max-reservable-bandwidth {
    type decimal64 {
        fraction-digits 2;
    }
    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 decimal64 {
            fraction-digits 2;
        }
        description "Unreserved bandwidth.";
    }
    description
        "List of unreserved bandwidths for different
        priorities.";

```

```

    }
}
leaf admin-group {
    type uint32;
    description "Administrative group/Resource class/Color.";
}
uses unknown-tlvs;
}
}

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

```

```

        description "OSPF inter-area route.";
    }
    enum external {
        value "5";
        description "OSPF External route.";
    }
    enum nssa {
        value "7";
        description "OSPF NSSA external route.";
    }
}
description "Route type.";
}
leaf flags {
    type bits {
        bit A {
            description
                "Attach flag.";
        }
        bit N {
            description
                "Node flag.";
        }
    }
}

```

```

    }
}
description "Flags.";
}
leaf prefix {
    type inet:ip-prefix;
    description "Address prefix.";
}
uses unknown-tlvs;
}
}

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

```

```

    }
  }
}

grouping ospfv3-lsa-options {
  description "OSPFv3 LSA options";
  leaf options {
    type bits {
      bit DC {
        description
          "When set, the router support demand circuits.";
      }
      bit R {
        description
          "When set, the originator is an active router.";
      }
      bit N {
        description
          "If set, the router is attached to an NSSA";
      }
      bit E {
        description
          "This bit describes the way AS-external-LSAs
            are flooded";
      }
      bit V6 {
        description
          "If clear, the router/link should be excluded
            from IPv6 routing calculaton";
      }
    }
  }
}

```

```

    }
  }
  mandatory true;
  description "OSPFv3 LSA options.";
}
}

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

  leaf prefix {

```



```

    type inet:ip-prefix;
    description
        "Prefix.";
}
leaf prefix-options {
    type bits {
        bit NU {
            description
                "When set, the prefix should be excluded
                from IPv6 unicast calculations.";
        }
        bit LA {
            description
                "When set, the prefix is actually an IPv6 interface
                address of the Advertising Router.";
        }
        bit P {
            description
                "When set, the NSSA area prefix should be
                readvertised by the translating NSSA area border.";
        }
        bit DN {
            description
                "When set, the inter-area-prefix-LSA or
                AS-external-LSA prefix has been advertised in a VPN
                environment.";
        }
    }
    mandatory true;
    description "Prefix options.";
}
}

grouping ospfv3-lsa-external {
    description
        "AS-External and NSSA LSA.";
}

```

```

leaf metric {
    type uint24;
    description "Metric";
}

```

```

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

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

uses ospfv3-lsa-prefix;

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

leaf external-route-tag {
  type uint32;
  description
    "Route tag.";
}

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

```

```

    }
}

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

uses ospfv3-lsa-options;

container links {
    description "All router link.";
    list link {
        description "Router LSA link.";
        leaf interface-id {
            type uint32;
            description "Interface ID.";
        }
    }
}

```

```
    }
    leaf neighbor-interface-id {
        type uint32;
        description "Neighbor Interface ID.";
    }
    leaf neighbor-router-id {
        type yang:dotted-quad;
        description "Neighbor Router ID.";
    }
    leaf type {
        type uint8;
        description "Link type.";
    }
    leaf metric {
        type uint16;
        description "Metric.";
    }
}
}
}
container network {
    when "../../header/type = 8194" { // 0x2002
        description
            "Only apply to network LSA.";
    }
    description "Network LSA.";

    uses ospfv3-lsa-options;

    container attached-routers {
        description "All attached routers.";
        leaf-list attached-router {
            type yang:dotted-quad;
            description
                "List of the routers attached to the network.";
        }
    }
}
container inter-area-prefix {
    when "../../header/type = 8195" { // 0x2003
        description
            "Only apply to inter-area-prefix LSA.";
```

```

}
leaf metric {
    type uint24;
    description "Metric";
}

```

```

    uses ospfv3-lsa-prefix;
    description "Inter-Area-Prefix LSA.";
}
container inter-area-router {
    when "../header/type = 8196" { // 0x2004
        description
            "Only apply to inter-area-router LSA.";
    }
    uses ospfv3-lsa-options;
    leaf metric {
        type uint24;
        description "Metric.";
    }
    leaf destination-router-id {
        type yang:dotted-quad;
        description
            "The Router ID of the router being described by the LSA.";
    }
    description "Inter-Area-Router LSA.";
}
container as-external {
    when "../header/type = 16389" { // 0x4005
        description
            "Only apply to as-external LSA.";
    }

    uses ospfv3-lsa-external;

    description "AS-External LSA.";
}
container nssa {
    when "../header/type = 8199" { // 0x2007
        description
            "Only apply to nssa LSA.";
    }
}

```

```

    uses ospfv3-lsa-external;

    description "NSSA LSA.";
}
container link {
    when ".././header/type = 8" { // 0x0008
        description
            "Only apply to link LSA.";
    }
    leaf rtr-priority {
        type uint8;
        description "Router Priority of the interface.";
    }
}

```

```

    uses ospfv3-lsa-options;

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

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

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

```

```

description "Intra-Area-Prefix LSA.";

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

leaf num-of-prefixes {
    type uint16;
    description "Number of prefixes.";
}

```

```

}
container prefixes {
    description "All prefixes in this LSA.";
    list prefix {
        description "List of prefixes in this LSA.";
        uses ospfv3-lsa-prefix;
        leaf metric {
            type uint24;
            description "Metric.";
        }
    }
}
}
}

grouping lsa-header {
    description
        "Common LSA for OSPFv2 and OSPFv3";
    leaf age {
        type uint16;
        mandatory true;
    }
}

```

```

    description "LSA age.";
}
leaf type {
    type uint16;
    mandatory true;
    description "LSA type.";
}
leaf adv-router {
    type yang:dotted-quad;
    mandatory true;
    description "LSA advertising router.";
}
leaf seq-num {
    type uint32;
    mandatory true;
    description "LSA sequence number.";
}
leaf checksum {
    type uint16;
    mandatory true;
    description "LSA checksum.";
}
leaf length {
    type uint16;
    mandatory true;
    description "LSA length.";
}
}

```

```

}

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

```



```

    "Decoded OSPFv2 LSA header data.";
  leaf option {
    type bits {
      bit DC {
        description
          "When set, the router support demand circuits.";
      }
      bit P {
        description
          "Only used in type-7 LSA. When set, the NSSA
           border router should translate the type-7 LSA
           to type-5 LSA.";
      }
      bit MC {
        description
          "When set, the router support MOSPF.";
      }
      bit E {
        description
          "This bit describes the way AS-external-LSAs
           are flooded";
      }
    }
    mandatory true;
    description "LSA option.";
  }
  leaf lsa-id {
    type inet:ipv4-address;
    mandatory true;
    description "LSA ID.";
  }

  leaf opaque-type {
    type uint8;

```

```

    description "Opaque type.";
  }

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

```

```

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

grouping ospfv3-lsa {
    description
        "Decoded OSPFv3 LSA.";
    container header {
        description
            "Decoded OSPFv3 LSA header data.";
        leaf lsa-id {
            type uint32;
            mandatory true;
            description "LSA ID.";
        }
        uses lsa-header;
    }
    container body {
        description
            "Decoded OSPF LSA body data.";
        uses ospfv3-lsa-body;
    }
}

grouping lsa-common {
    description
        "Common field for OSPF LSA represenation.";
    leaf decoded-completed {
        type boolean;
        description
            "The OSPF LSA body is fully decoded.";
    }
    leaf raw-data {
        type yang:hex-string;
        description

```

```

        "The complete LSA in network byte
        order as received/sent over the wire.";
    }
}

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

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

grouping af-area-config {
    description
        "OSPF address-family specific area config state.";
    container ranges {
        description "Container for summary ranges";
        list range {
            key "prefix";

```

```
    description
      "Summarize routes matching address/mask (border
        routers only)";
    leaf prefix {
      type inet:ip-prefix;
      description
        "IPv4 or IPv6 prefix";
    }
    leaf advertise {
      type boolean;
      description
        "Advertise or hide.";
    }
    leaf cost {
      type uint24 {
        range "0..16777214";
      }
      description
        "Cost of summary route.";
    }
  }
}

grouping area-config {
  description
    "OSPF area config state.";
  leaf area-type {
    type identityref {
      base area-type;
    }
    default normal;
    description
      "Area type.";
  }

  leaf summary {
    when "../area-type = 'stub' or ../area-type = 'nssa'" {
      description
        "Summary generation valid for stub/NSSA area.";
    }
    type boolean;
    description
```

```

        "Enable/Disable summary generation to the stub or
        NSSA area.";
    }

    leaf default-cost {

```

```

        when "../area-type = 'stub' or ../area-type = 'nssa'" {
            description
                "Default cost for LSA advertised into stub or
                NSSA area.";
        }
        type uint32 {
            range "1..16777215";
        }
        description
            "Set the summary default-cost for a stub or NSSA area.";
    }
}

grouping local-rib {
    description "Local-rib grouping.";
    container local-rib {
        description "Local-rib.";
        list route {
            key "prefix";
            description "Routes";
            leaf prefix {
                type inet:ip-prefix;
                description "Destination prefix.";
            }
            container next-hops {
                description "All next hops for the route.";
                list next-hop {
                    key "next-hop";
                    description "List of next hop for the route";
                    leaf outgoing-interface {
                        type if:interface-ref;
                        description
                            "Name of the outgoing interface.";
                    }
                }
                leaf next-hop {
                    type inet:ip-address;

```

```

        description "Nexthop address.";
    }
}
}
leaf metric {
    type uint32;
    description "Metric for this route.";
}
leaf route-type {
    type route-type;
    description "Route type for this route.";
}

```

```

    leaf route-tag {
        type uint32;
        description "Route tag for this route.";
    }
}
}
}

```

```

grouping instance-config {
    description
        "OSPF instance config state.";
    leaf router-id {
        if-feature router-id;
        type yang:dotted-quad;
        description
            "Defined in RFC 2328. A 32-bit number
            that uniquely identifies the router.";
    }
}

```

```

container admin-distance {
    description "Admin distance config state.";
    choice scope {
        description
            "Options for expressing admin distance
            as single or multiple values.";
        case single-value {
            leaf all {
                type uint8;
                description

```

```

        "Admin distance for intra-area, inter-area and
        external route.";
    }
}
case multi-values {
    choice granularity {
        description
            "Options for expressing admin distance
            for intra-area and inter-area route.";
        case detail {
            leaf intra-area {
                type uint8;
                description
                    "Admin distance for intra-area route.";
            }
            leaf inter-area {
                type uint8;
                description
                    "Admin distance for inter-area route.";
            }
        }
    }
}

```

```

    }
}
case coarse {
    leaf internal {
        type uint8;
        description
            "Admin distance for both intra-area and
            inter-area route.";
    }
}
}
leaf external {
    type uint8;
    description
        "Admin distance for both external route.";
}
}
}
}

container nsr {
    if-feature nsr;
}

```

```

description
  "NSR config state.";
leaf enable {
  type boolean;
  description
    "Enable/Disable NSR.";
}
}

container graceful-restart {
  if-feature graceful-restart;
  description
    "Graceful restart config state.";
  leaf enable {
    type boolean;
    description
      "Enable/Disable graceful restart as defined in RFC 3623.";
  }
  leaf helper-enable {
    type boolean;
    description
      "Enable RestartHelperSupport in RFC 3623 Section B.2.";
  }
  leaf restart-interval {
    type uint16 {
      range "1..1800"; // Range is defined in RFC 3623.
    }
  }
}

```

```

    }
    units seconds;
    default "120"; // Default is defined in RFC 3623.
    description
      "RestartInterval option in RFC 3623 Section B.1.";
  }
  leaf helper-strict-lsa-checking {
    type boolean;
    description
      "RestartHelperStrictLSAChecking option in RFC 3623
      Section B.2.";
  }
}

leaf enable {

```



```

    if-feature admin-control;
    type boolean;
    default true;
    description
        "Enable/Disable the protocol.";
}

container auto-cost {
    if-feature auto-cost;
    description
        "Auto cost config state.";
    leaf enable {
        type boolean;
        description
            "Enable/Disable 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 in term of Mbits";
    }
}

container spf-control {
    leaf paths {
        if-feature max-ecmp;
        type uint16 {

```

```

        range "1..32";
    }
    description
        "Maximum number of ECMP paths.";
    }
    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 will receive.";
  }
  description "Database maintenance control.";
}

container stub-router {
  if-feature stub-router;
  description "Set maximum metric configuration";

  choice trigger {
    description
      "Specific different triggers to enable stub router.";
    container always {
      presence
        "Enables maximum metric for non-stub router link";
      description "Set maximum metric always configuration";
    }
  }
}

container mpls {
  description
    "OSPF MPLS config state.";
  container te-rid {
    if-feature te-rid;
    description
      "Traffic Engineering stable IP address for system.";
    choice source {
      description
        "Different options for specifying TE router ID.";
      case interface {
        leaf interface {
          type if:interface-ref;

```

```

        "Take the interface's IPv4 address as TE
        router ID.";
    }
}
case explicit {
    leaf router-id {
        type inet:ipv4-address;
        description
            "Explicitly configure the TE 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.";
    }
    leaf autoconfig {
        if-feature ldp-igp-autoconfig;
        type boolean;
        description
            "Enable LDP IGP interface auto-configuration.";
    }
}
}
uses instance-fast-reroute-config;
}

grouping interface-operation {
    description
        "OSPF interface operation state.";
    reference "RFC2328 Section 9";
    uses interface-config;

    leaf state {
        type if-state-type;
        description "Interface state.";
    }

    leaf hello-timer {
        type uint32;
        units "milliseconds";
    }
}

```

```
    description "Hello timer.";
  }

  leaf wait-timer {
    type uint32;
    units "milliseconds";
    description "Wait timer.";
  }

  leaf dr {
    type inet:ipv4-address;
    description "DR.";
  }

  leaf bdr {
    type inet:ipv4-address;
    description "BDR.";
  }

  container statistics {
    description "Per interface statistics";
    uses interface-stat;
  }
} // interface-operation

grouping neighbor-operation {
  description
    "OSPF neighbor operation data.";

  leaf address {
    type inet:ip-address;
    description
      "Neighbor address.";
  }

  leaf dr {
    type yang:dotted-quad;
    description
      "Designated Router.";
  }

  leaf bdr {
    type yang:dotted-quad;
    description
      "Backup Designated Router.";
  }

  leaf state {
    type nbr-state-type;
```

```
description
  "OSPF neighbor state.";
```

```
    }
    container statistics {
      description "Per neighbor statistics";
      uses neighbor-stat;
    }
  }

  grouping instance-operation {
    description
      "OSPF Address Family operation state.";
    leaf router-id {
      type yang:dotted-quad;
      description
        "Defined in RFC 2328. A 32-bit number
        that uniquely identifies the router.";
    }

    uses local-rib;

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

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

    container ospf {
      description
        "OSPF.";

      container all-instances-inherit {
```

```

if-feature instance-inheritance;
description
    "Inheritance support to all instances.";
container area {
    description
        "Area config to be inherited by all areas in
        all instances.";
}
container interface {

```

```

    description
        "Interface config to be inherited by all interfaces
        in all instances.";
}
}

leaf operation-mode {
    type identityref {
        base operation-mode;
    }
    default ospf:ships-in-the-night;
    description
        "OSPF operation mode.";
}

list instance {
    key "af";
    description
        "An OSPF routing protocol instance.";

    leaf af {
        type identityref {
            base rt:address-family;
        }
        description
            "Address-family of the instance.";
    }

    uses instance-config;

    container all-areas-inherit {
        if-feature area-inheritance;
    }
}

```

```

description
  "Inheritance for all areas.";
container area {
  description
    "Area config to be inherited by all areas.";
}
container interface {
  description
    "Interface config to be inherited by all interfaces
    in all areas.";
}
}

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 af-area-config {
  when "../..operation-mode = "
    + "'ospf:ships-in-the-night'" {
    description
      "Ships in the night configuration.";
  }
}

container all-interfaces-inherit {
  if-feature interface-inheritance;
  description
    "Inheritance for all interfaces";
  container interface {
    description
      "Interface config to be inherited by all
      interfaces.";

```

```

    }
}

container virtual-links {
  description "All virtual links.";
  list virtual-link {
    when "../..area-id != '0' and "
      + "../..area-type = 'normal'" {
      description
        "Transit area must be non-backbone normal area.";
    }
    key "router-id";
    description
      "OSPF virtual link";
    leaf router-id {
      type yang:dotted-quad;
      description
        "Virtual link router ID.";
    }
  }

  uses interface-common-config;
}

```

```

container sham-links {
  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 end-point.";
    }
    leaf remote-id {
      type inet:ip-address;
      description
        "Address of the remote end-point.";
    }
  }

  uses interface-common-config;
}

```



```

    }
    container interfaces {
        description "All interfaces.";
        list interface {
            key "interface";
            description
                "List of OSPF interfaces.";
            leaf interface {
                type if:interface-ref;
                description
                    "Interface.";
            }
            uses interface-config;
        } // list of interfaces
    } // list of areas
} // list of instance
} // container ospf
}

augment "/rt:routing/rt:routing-instance/rt:routing-protocols/"
+ "rt:routing-protocol/ospf:ospf/ospf:instance" {
    when "../..//rt:type = 'ospf:ospfv2' or
        ../..//rt:type = 'ospf:ospfv3'" {
        description
            "This augment is only valid for OSPF
            (type 'ospfv2' or 'ospfv3').";
    }
    if-feature multi-topology;
}

```

```

description
    "OSPF multi-topology routing-protocol augmentation.";
container topologies {
    description "All topologies.";
    list topology {
        // Topology must be in the same routing-instance
        // and of same AF as the container.
        key "name";
        description "OSPF topology.";
        leaf name {
            type leafref {

```



```

    key "name";
    description "OSPF interface topology.";
    leaf name {
        type leafref {
            path "../..../..../..../..../..../..../..../..../"
                + "rt:ribs/rt:rib/rt:name";
        }
        description
            "One of the topology enabled on this interface.";
    }
    leaf cost {
        type uint32;
        description
            "Interface cost for this topology.";
    }
}
}
}

augment "/rt:routing-state/rt:routing-instance/"
    + "rt:routing-protocols/rt:routing-protocol" {
    when "rt:type = 'ospf:ospfv2' or rt:type = 'ospf:ospfv3'" {
        description
            "This augment is only valid for a routing protocol instance
            of type 'ospfv2' or 'ospfv3'.";
    }
    description
        "OSPF operation state.";
    container ospf {
        description "OSPF";

        leaf operation-mode {
            type identityref {
                base operation-mode;
            }
            description
                "OSPF operation mode.";
        }

        list instance {
            key "af";
            description
                "An OSPF routing protocol instance.";
        }
    }
}

```

```
leaf af {
  type identityref {
    base rt:address-family;
  }
  description
    "Address-family of the instance.";
}

uses instance-operation;

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.";
    }

    container statistics {
      description "Per area statistics";
      uses area-stat;
    }

    container virtual-links {
      description "All virtual links.";
      list virtual-link {
        description
          "OSPF virtual link";
        leaf router-id {
          type yang:dotted-quad;
          description
            "Virtual link router ID.";
        }
      }

      uses interface-operation;
    }
  }

  container sham-links {
    description "All sham links.";
    list sham-link {
      description
        "OSPF sham link";
      leaf local-id {
        type inet:ip-address;
        description

```

"Address of the local end-point.";

```
    }
    leaf remote-id {
      type inet:ip-address;
      description
        "Address of the remote end-point.";
    }
    uses interface-operation;
  }
}

container interfaces {
  description "All interfaces in the area.";
  list interface {
    key "interface";
    description
      "List of OSPF interfaces.";
    leaf interface {
      // Should it refer to config state leaf?
      type if:interface-ref;
      description "Interface.";
    }
  }

  uses interface-operation;

  container neighbors {
    description "All neighbors on the interface.";
    list neighbor {
      key "neighbor-router-id";
      description
        "List of OSPF neighbors.";
      leaf neighbor-router-id {
        type yang:dotted-quad;
        description
          "Neighbor router ID.";
      }

      uses neighbor-operation;
    } // list of OSPF neighbors
  }
}
```

```

container database {
  description "Link scope LSA database.";
  list link-scope-lsa-type {
    key "lsa-type";
    description
      "List OSPF link scope LSA databases.";
    leaf lsa-type {
      type uint16;
    }
  }
}

```

```

    description "OSPF link scope LSA type.";
  }
  container link-scope-lsas {
    description
      "All link scope LSAs of this LSA type.";
    list link-scope-lsa {
      key "lsa-id adv-router";
      description "List of OSPF link scope LSAs";
      uses lsa-key;
      uses lsa {
        refine "version/ospfv2/ospfv2" {
          must "../..../..../..../..../..../..../..../..../"
            + "rt:type = 'ospfv2'" {
            description "OSPFv2 LSA.";
          }
        }
        refine "version/ospfv3/ospfv3" {
          must "../..../..../..../..../..../..../..../..../"
            + "rt:type = 'ospfv3'" {
            description "OSPFv3 LSA.";
          }
        }
      }
    }
  }
} // list link-scope-lsas
} // list of OSPF interfaces
}

container database {
  description "Area scope LSA database.";
  list area-scope-lsa-type {

```



```

    container as-scope-lsas {
      description "All AS scope of LSA of this LSA type.";
      list as-scope-lsa {
        key "lsa-id adv-router";
        description "List of OSPF area scope LSAs";
        uses lsa-key;
        uses lsa {
          refine "version/ospfv2/ospfv2" {
            must "../.../rt:type = 'ospfv2'" {
              description "OSPFv2 LSA.";
            }
          }
          refine "version/ospfv3/ospfv3" {
            must "../.../rt:type = 'ospfv3'" {
              description "OSPFv3 LSA.";
            }
          }
        }
      }
    }
  } // list as-scope-lsas
}

```

```

    } // list of instances
  } // container ospf
}

augment "/rt:routing-state/rt:routing-instance/"
  + "rt:routing-protocols/rt:routing-protocol/"
  + "ospf:ospf/ospf:instance" {
  when "../.../rt:type = 'ospf:ospfv2'" {
    description
      "This augment is only valid for OSPFv2.";
  }
  if-feature multi-topology;
  description
    "OSPF multi-topology routing-protocol augmentation.";
  container topologies {
    description "All topologies.";
    list topology {
      // Topology must be in the same routing-instance
      // and of same AF as the container.
    }
  }
}

```



```

    key "name";
    description "OSPF topology.";
    leaf name {
        type leafref {
            path "../..../..../..../.."
              + "rt:ribs/rt:rib/rt:name";
        }
        description "RIB";
    }

    uses local-rib;

    container areas {
        description "All areas in the topology.";
        list area {
            key "area-id";
            description
                "List of ospf areas";
            leaf area-id {
                type area-id-type;
                description
                    "Area ID.";
            }
        }
    }
}

```

```

augment "/rt:routing-state/rt:routing-instance/"
  + "rt:routing-protocols/rt:routing-protocol/"
  + "ospf:ospf/ospf:instance/ospf:areas/ospf:area/"
  + "ospf:interfaces/ospf:interface" {
    when "../..../..../..../rt:type = 'ospf:ospfv2'" {
        description
            "This augment is only valid for OSPFv2.";
    }
    if-feature ospf:multi-topology;
    description "OSPF multi-topology interface augmentation.";
    container topologies {
        description "All topologies.";
    }
}

```

```

list topology {
  key "name";
  description "OSPF interface topology.";
  leaf name {
    type leafref {
      path "../..../..../..../..../..../..../..../..../"
        + "rt:ribs/rt:rib/rt:name";
    }
    description
      "One of the topology enabled on this interface.";
  }
}
}
}

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-state/rt:routing-instance/"
  + "rt:ribs/rt:rib/rt:routes/rt:route" {
  when "rt:source-protocol = 'ospf:ospfv2' or "

```

```

  + "rt:source-protocol = 'ospf:ospfv3'" {
  description
    "This augment is only valid for a routes whose source
      protocol is OSPF.";
  }

```

```

    description
      "OSPF-specific route attributes.";
    uses route-content;
  }

  identity if-link-type {
    description "Base identity for OSPF interface link type.";
  }

  identity if-link-type-normal {
    base if-link-type;
    description "OSPF interface link type normal.";
  }

  identity if-link-type-virtual-link {
    base if-link-type;
    description "OSPF interface link type virtual link.";
  }

  identity if-link-type-sham-link {
    base if-link-type;
    description "OSPF interface link type sham link.";
  }

  grouping notification-instance-hdr {
    description
      "This group describes common instance specific
      data for notifications.";

    leaf routing-instance {
      type rt:routing-instance-ref;
      description
        "Describe the routing instance.";
    }

    leaf routing-protocol-type {
      type leafref {
        path "/rt:routing/rt:routing-instance"
          + "[rt:name=current()]/../routing-instance)/"
          + "rt:routing-protocols/rt:routing-protocol/"
          + "rt:type";
      }
      must ". = 'ospf:ospfv2' or . = 'ospf:ospfv3'" {

```

```

        description "Only interested in OSPF routing protocol";
    }
    description
        "Describes the type of the OSPF routing protocol.";
}

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

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

notification if-state-change {
    uses notification-instance-hdr;

    leaf link-type {
        type identityref {
            base if-link-type;
        }
        description "Type of OSPF interface.";
    }
}

container interface {
    description "Normal interface.";
    leaf interface {
        type if:interface-ref;
        description "Interface.";
    }
}

```

```
}
container virtual-link {
  description "virtual-link.";
  leaf area-id {
    type uint32;
    description "Area ID.";
  }
  leaf neighbor-router-id {
    type yang:dotted-quad;
    description "Neighbor router id.";
  }
}
container sham-link {
  description "sham-link.";
  leaf area-id {
    type uint32;
    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.";
  }
}

leaf state {
  type if-state-type;
  description "Interface state.";
}

description
  "This notification is sent when interface
  state change is detected.";
}

notification if-config-error {
  uses notification-instance-hdr;
```

```

leaf link-type {
  type identityref {
    base if-link-type;
  }
  description "Type of OSPF interface.";
}

```

```

container interface {
  description "Normal interface.";
  leaf interface {
    type if:interface-ref;
    description "Interface.";
  }
  leaf packet-source {
    type yang:dotted-quad;
    description "Source address.";
  }
}
container virtual-link {
  description "virtual-link.";
  leaf area-id {
    type uint32;
    description "Area ID.";
  }
  leaf neighbor-router-id {
    type yang:dotted-quad;
    description "Neighbor router id.";
  }
}
container sham-link {
  description "sham-link.";
  leaf area-id {
    type uint32;
    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.";
    }
}

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

leaf error {
    type enumeration {
        enum "badVersion" {

```

```

        description "Bad version.";
    }
    enum "areaMismatch" {
        description "Area mismatch.";
    }
    enum "unknownNbmaNbr" {
        description "Unknown NBMA neighbor.";
    }
    enum "unknownVirtualNbr" {
        description "Unknown virtual link neighbor.";
    }
    enum "authTypeMismatch" {
        description "Auth type mismatch.";
    }
    enum "authFailure" {
        description "Auth failure.";
    }
    enum "netMaskMismatch" {
        description "Network mask mismatch.";
    }
    enum "helloIntervalMismatch" {
        description "Hello interval mismatch.";
    }
    enum "deadIntervalMismatch" {
        description "Dead interval mismatch.";
    }
    enum "optionMismatch" {
        description "Option mismatch.";
    }

```

```

    }
    enum "mtuMismatch" {
        description "MTU mismatch.";
    }
    enum "duplicateRouterId" {
        description "Duplicate router ID.";
    }
    enum "noError" {
        description "No error.";
    }
}
description "Error code.";
}
description
    "This notification is sent when interface
    config error is detected.";
}

notification nbr-state-change {
    uses notification-instance-hdr;

```

```

leaf link-type {
    type identityref {
        base if-link-type;
    }
    description "Type of OSPF interface.";
}

container interface {
    description "Normal interface.";
    leaf interface {
        type if:interface-ref;
        description "Interface.";
    }
    leaf neighbor-router-id {
        type yang:dotted-quad;
        description "Neighbor router id.";
    }
    leaf neighbor-ip-addr {
        type yang:dotted-quad;
        description "Neighbor address.";
    }
}

```



```

}
container virtual-link {
  description "virtual-link.";
  leaf area-id {
    type uint32;
    description "Area ID.";
  }
  leaf neighbor-router-id {
    type yang:dotted-quad;
    description "Neighbor router id.";
  }
}
container sham-link {
  description "sham-link.";
  leaf area-id {
    type uint32;
    description "Area ID.";
  }
  leaf local-ip-addr {
    type inet:ip-address;
    description "Sham link local address.";
  }
  leaf neighbor-router-id {
    type yang:dotted-quad;
    description "Neighbor router id.";
  }
  leaf neighbor-ip-addr {

```

```

    type yang:dotted-quad;
    description "Neighbor address.";
  }
}

leaf state {
  type nbr-state-type;
  description "Neighbor state.";
}

description
  "This notification is sent when neighbor
  state change is detected.";
}

```

```

notification nbr-restart-helper-status-change {
    uses notification-instance-hdr;

    leaf link-type {
        type identityref {
            base if-link-type;
        }
        description "Type of OSPF interface.";
    }

    container interface {
        description "Normal interface.";
        leaf interface {
            type if:interface-ref;
            description "Interface.";
        }
        leaf neighbor-router-id {
            type yang:dotted-quad;
            description "Neighbor router id.";
        }
        leaf neighbor-ip-addr {
            type yang:dotted-quad;
            description "Neighbor address.";
        }
    }
    container virtual-link {
        description "virtual-link.";
        leaf area-id {
            type uint32;
            description "Area ID.";
        }
        leaf neighbor-router-id {
            type yang:dotted-quad;

```

```

        description "Neighbor router id.";
    }
}

leaf status {
    type restart-helper-status-type;
    description "Restart helper status.";

```

```

}

leaf age {
    type uint32;
    units seconds;
    description
        "Remaining time in current OSPF graceful restart
        interval, if the router is acting as a restart
        helper for the neighbor.";
}

leaf exit-reason {
    type restart-exit-reason-type;
    description
        "Restart helper exit reason.";
}
description
    "This notification is sent when neighbor restart
    helper status change is detected.";
}

notification rx-bad-packet {
    uses notification-instance-hdr;

    leaf link-type {
        type identityref {
            base if-link-type;
        }
        description "Type of OSPF interface.";
    }

    container interface {
        description "Normal interface.";
        leaf interface {
            type if:interface-ref;
            description "Interface.";
        }
        leaf packet-source {
            type yang:dotted-quad;
            description "Source address.";
        }
    }
}

```

```

}
container virtual-link {
    description "virtual-link.";
    leaf area-id {
        type uint32;
        description "Area ID.";
    }
    leaf neighbor-router-id {
        type yang:dotted-quad;
        description "Neighbor router id.";
    }
}

container sham-link {
    description "sham-link.";
    leaf area-id {
        type uint32;
        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.";
    }
}

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

description
    "This notification is sent when an OSPF packet
    has been received on a interface that cannot be parsed.";
}

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.";
    }
}

```

Internet-Draft

OSPF Yang Data Model

March 2015

```
    }

    description
      "This notification is sent when the number of LSAs
       in the router's link state database has exceeded
       ninety percent of the ext-lsdb-limit.";
  }

  notification lsdb-overflow {
    uses notification-instance-hdr;

    leaf ext-lsdb-limit {
      type uint32;
      description
        "The maximum number of non-default AS-external LSAs
         entries that can be stored in the link state database.";
    }

    description
      "This notification is sent when the number of LSAs
       in the router's link state database has exceeded
       ext-lsdb-limit.";
  }

  notification nssa-translator-status-change {
    uses notification-instance-hdr;

    leaf area-id {
      type uint32;
      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 ability to translate OSPF NSSA LSAs
       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>

```

[5.](#) OSPF Segment Routing Yang Module

```

<CODE BEGINS> file "ietf-ospf-sr@2015-07-06.yang"
module ietf-ospf-sr {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-sr";

    prefix ospf-sr;

```

```

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

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

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

```

Yeung, et al.

Expires September 2, 2015

[Page 90]

Internet-Draft

OSPF Yang Data Model

March 2015

```

    prefix "sr";
}
import ietf-ospf {
    prefix "ospf";
}

organization
    "Cisco Systems
    170 West Tasman Drive
    San Jose, CA 95134-1706
    USA";

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

    WG Chair: Acee Lindem
               <mailto:acee@cisco.com>

    WG Chair: Abhay Roy
               <mailto:akr@cisco.com>

    Editor:    Derek Yeung
               <mailto:myeung@cisco.com>
    Author:    Derek Yeung
               <mailto:myeung@cisco.com>
    Author:    Yingzhen Qu
               <mailto:yiqu@cisco.com>
    Author:    Jeffrey Zhang

```

Author: <mailto:zzhang@juniper.net>
Ing-Wher Chen
<mailto:ing-wher.chen@ericsson.com>
Author: Greg Hankins
<mailto:greg.hankins@alcatel-lucent.com>;

description

"This YANG module defines the generic configuration data for OSPF, which is common across all of the vendor implementations of the protocol. 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.

Terms and Acronyms

OSPF (ospf): Open Shortest Path First

IP (ip): Internet Protocol

IPv4 (ipv4):Internet Protocol Version 4

IPv6 (ipv6): Internet Protocol Version 6

MTU (mtu) Maximum Transmission Unit
";

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

```
feature ti-lfa {  
  description  
    "Enhance IPFRR with ti-lfa support";  
}
```

```
/* Configuration */
```

```
augment "/rt:routing/rt:routing-instance/rt:routing-protocols"
```



```

    + "/rt:routing-protocol/ospf:ospf/ospf:instance" {
when "../..//rt:type = 'ospf:ospfv2' or "
    + "../..//rt:type = 'ospf:ospfv3'" {
    description
    "This augment OSPF routing protocol when used";
}
description
    "This augments OSPF protocol configuration
    with segment routing.";
uses sr:controlplane-cfg;
}

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

uses sr:igp-interface-cfg;
}

```

```

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

container ti-lfa {
    if-feature ti-lfa;
    leaf enable {

```

```

        type boolean;
        description
            "Enables TI-LFA computation.";
    }
    description
        "TILFA configuration.";
}

/* Operational states */

augment "/rt:routing-state/rt:routing-instance/"
    + "rt:routing-protocols/rt:routing-protocol/"
    + "ospf:ospf/ospf:instance" {
    when "../..//rt:type = 'ospf:ospfv2' or "
        + "../..//rt:type = 'ospf:ospfv3'" {
        description
            "This augment OSPF routing protocol when used";
    }
    description
        "This augments OSPF protocol configuration
        with segment routing.";

    uses sr:controlplane-cfg;
}

augment "/rt:routing-state/rt:routing-instance/"
    + "rt:routing-protocols/rt:routing-protocol/"
    + "ospf:ospf/ospf:instance/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface" {
    when "../..//../..//../..//rt:type = 'ospf:ospfv2' or "
        + "../..//../..//../..//rt:type = 'ospf:ospfv3'" {
        description

```

```

        "This augment OSPF routing protocol when used";
    }
    description
        "This augments OSPF protocol configuration
        with segment routing.";

    uses sr:igp-interface-cfg;
}

```

```

/* Database */

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

grouping prefix-sid-sub-tlvs {
  description "Prefix SID sub-tlvs";
  container prefix-sid-sub-tlvs {
    description "prefix sid sub-tlv.";
    list prefix-sid-sub-tlv {
      description "prefix sid sub-tlv.";
      leaf flags {
        type bits {
          bit NP {
            description
              "No-PHP flag.";
          }
          bit M {
            description
              "Mapping server flag.";
          }
          bit E {
            description
              "Explicit-NULL flag.";
          }
          bit V {
            description
              "Value/Index flag.";
          }
          bit L {

```

```

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

grouping sid-binding-sub-tlvs {
    description "SID/Label binding sub-tlv grouping.";
    container sid-binding-sub-tlvs {
        description "sid binding sub-tlvs.";
        list sid-binding-sub-tlv {
            description "sid binding sub-tlv.";
            leaf flags {
                type bits {
                    bit M {
                        description
                            "Mirroring context flag.";
                    }
                }
            }
            description "Flags.";
        }
        leaf mt-id {
            type uint8;
            description "Multi-topology ID.";
        }
        leaf weight {
            type uint8;
            description "Weight for load-balancing purposes.";
        }
    }
}

```

```
uses sid-sub-tlv;

container ero-metric-sub-tlv {
  description "The cost of an ERO path.";
  leaf metric {
    type uint32;
    description "The aggregate IGP or TE path cost.";
  }
}

container ipv4-ero-sub-tlv {
  description
    "The ipv4 ERO sub-tlv describes a path segment
      using ipv4 address.";
  leaf flags {
    type bits {
      bit L {
        description
          "If set, then the segment path is designated as
            'loose'. Otherwise as 'strict'.";
      }
    }
    description "Flags.";
  }
  leaf ipv4-address {
    type inet:ipv4-address;
    description "The address of the explicit route hop.";
  }
}

container unnumbered-ero-sub-tlv {
  description "The unnumbered interface ID ERO sub-tlv
    describes a path segment that includes an
    unnumbered interface.";
  leaf flags {
    type bits {
      bit L {
        description
          "If set, then the segment path is designated as
            'loose'. Otherwise as 'strict'.";
      }
    }
    description "Flags.";
  }
  leaf router-id {
    type yang:dotted-quad;
    description "Router-id of the next-hop.";
  }
}
```

```
}
leaf interface-id {
```

```
    type uint32;
    description
        "The identifier assigned to the link by the
         router specified by the router-id.";
}
}
container ipv4-backup-ero-sub-tlv {
    description
        "The ipv4 backup ERO sub-tlv describes a path
         segment using ipv4 address.";
    leaf flags {
        type bits {
            bit L {
                description
                    "If set, then the segment path is designated as
                     'loose'. Otherwise as 'strict'.";
            }
        }
        description "Flags.";
    }
    leaf ipv4-address {
        type inet:ipv4-address;
        description "The address of the explicit route hop.";
    }
}
container unnumbered-backup-ero-sub-tlv {
    description "The unnumbered interface ID backup ERO sub-tlv
         describes a path segment that includes an
         unnumbered interface.";
    leaf flags {
        type bits {
            bit L {
                description
                    "If set, then the segment path is designated as
                     'loose'. Otherwise as 'strict'.";
            }
        }
        description "Flags.";
    }
}
```

```

leaf router-id {
    type yang:dotted-quad;
    description "Router-id of the next-hop.";
}
leaf interface-id {
    type uint32;
    description
        "The identifier assigned to the link by the
         router specified by the router-id.";
}

```

```

    }
  }
}
}
}

```

```

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

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

```

```

        uses sid-binding-sub-tlvs;
        uses ospf:unknown-tlvs;
    }
}

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

```

```

}

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

augment "/rt:routing-state/rt:routing-instance/"
    + "rt:routing-protocols/rt:routing-protocol/"
    + "ospf:ospf/ospf:instance/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface/ospf:database/"
    + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
    + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-tlvs/ospf:extended-prefix-tlv" {

```



```

        bit L {
            description
                "Local/Global flag.";
        }
        bit S {
            description
                "Set flag.";
        }
    }
    description "Flags.";
}
leaf mt-id {
    type uint8;
    description "Multi-topology ID.";
}
leaf weight {
    type uint8;
    description "Weight used for load-balancing.";
}
leaf sid {
    type uint32;
    description "sid index/label.";
}
}
}

```

```

container lan-adj-sid-sub-tlvs {
    description "Lan adj-sid is optional sub-tlv.";
    list lan-adj-sid-sub-tlv {
        description "List of lan-adj-sid sub-tlv.";
        leaf flags {
            type bits {
                bit B {
                    description
                        "Backup flag.";
                }
                bit V {

```

```

        description
            "Value/Index flag.";
    }
    bit L {

```

```

        description
            "Local/Global flag.";
    }
    bit S {
        description
            "Set flag.";
    }
}
description "Flags.";
}
leaf mt-id {
    type uint8;
    description "Multi-topology ID.";
}
leaf weight {
    type uint8;
    description "Weight used for load-balancing.";
}
leaf neighbor-router-id {
    type yang:dotted-quad;
    description "Neighbor router ID.";
}
leaf sid {
    type uint32;
    description "sid index/label.";
}
}
}
}

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

description

```

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

    uses extended-prefix-range-tlvs;
    uses sr-algorithm-tlv;
    uses sid-range-tlvs;
}

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

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

    uses extended-prefix-range-tlvs;
    uses sr-algorithm-tlv;
    uses sid-range-tlvs;
}

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

  description
    "SR specific TLVs for OSPFv2 type 11 opaque LSA.";

    uses extended-prefix-range-tlvs;
    uses sr-algorithm-tlv;
    uses sid-range-tlvs;
}
}
```

[6.](#) Security Considerations

The data model defined does not create any security implications.

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

[7.](#) Acknowledgements

The authors wish to thank Acee Lindem, Yi Yang, Alexander Clemm, Gaurav Gupta, Ing-Wher Chen, Ladislav Lhotka, Stephane Litkowski, Greg Hankins and Manish Gupta for their thorough reviews and helpful comments.

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

[8.](#) References

[8.1.](#) Normative References

- [RFC1793] Moy, J., "Extending OSPF to Support Demand Circuits", [RFC 1793](#), April 1995.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, [RFC 2328](#), April 1998.
- [RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", [RFC 3101](#), January 2003.
- [RFC3623] Moy, J., Pillay-Esnault, P., and A. Lindem, "Graceful OSPF Restart", [RFC 3623](#), November 2003.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", [RFC 3630](#), September 2003.
- [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](#), June 2006.

[RFC4750] Joyal, D., Galecki, P., Giacalone, S., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", [RFC 4750](#), December 2006.

[RFC5187] Pillay-Esnault, P. and A. Lindem, "OSPFv3 Graceful Restart", [RFC 5187](#), June 2008.

Yeung, et al.

Expires September 2, 2015

[Page 104]

Internet-Draft

OSPF Yang Data Model

March 2015

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", [RFC 5340](#), July 2008.

[RFC5643] Joyal, D. and V. Manral, "Management Information Base for OSPFv3", [RFC 5643](#), August 2009.

[RFC5838] Lindem, A., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", [RFC 5838](#), April 2010.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), May 2014.

[8.2](#). Informative References

[I-D.acee-rtg-yang-key-chain]

Lindem, A., Qu, Y., Yeung, D., Chen, H., Zhang, J., and Y. Yang, "Key Chain YANG Data Model", [draft-acee-rtg-yang-key-chain-07](#) (work in progress), July 2015.

[I-D.ietf-netmod-routing-cfg]

Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", [draft-ietf-netmod-routing-cfg-19](#) (work in progress), May 2015.

[I-D.litkowski-spring-sr-yang]

Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", [draft-litkowski-spring-sr-yang-01](#) (work in progress), June 2015.

Authors' Addresses

Derek Yeung
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

E-Mail: myeung@cisco.com

Yeung, et al.

Expires September 2, 2015

[Page 105]

Internet-Draft

OSPF Yang Data Model

March 2015

Yingzhen Qu
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

E-Mail: yiqu@cisco.com

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

E-Mail: zzhang@juniper.net

Dean Bogdanovic
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

E-Mail: deanb@juniper.net

Kiran Agrahara Sreenivasa
Brocade Communications System
9442 Capital of Texas Hwy North
Arboretum Plaza One, Suite 500
Austin, TX 78759
USA

EMail: kkoushik@brocade.com